Masterarbeit

VISUALISIERUNG UND VIRTUELLE STEUERUNG EINES DIGITAL TWIN IM ELEKTRO-ENERGIETECHNIK-LABOR DER FH CAMPUS 02

ausgeführt am

Fachhochschul-Masterstudiengang
Automatisierungstechnik-Wirtschaft

von
Barbara Semler, BSc
1810322024

betreut und begutachtet von
FH-Prof. Dipl.-Ing. Dieter Lutzmayr

Graz, im Dezember 2019 Unterschrift

EHRENWÖRTLICHE ERKLÄRUNG

Ich erkläre ehrenwörtlich, dass ich die vorliegende Arbeit selbstständig und ohne fremde Hilfe verfasst, andere als die angegebenen Quellen nicht benützt und die benutzten Quellen wörtlich zitiert sowie inhaltlich entnommene Stellen als solche kenntlich gemacht habe.

.....

Unterschrift

DANKSAGUNG

Ich möchte mich bei all jenen bedanken, die direkt oder indirekt zur Entstehung dieser Masterarbeit in Form von fachlicher und moralischer Unterstützung beigetragen haben.

Herzlicher Dank gilt meiner Familie, die mich die gesamte Studienzeit hindurch mit enormer Geduld und Verständnis in all meinen Entscheidungen und bei der Erreichung meiner Ziele unterstützt und mir die Kraft und die Motivation gegeben hat, alle Hürden und Herausforderungen, die sich mir im Laufe dieser Zeit in den Weg gestellt haben, zu bewältigen.

Schließlich danke ich meinen Freunden während der Studienzeit, die mich über viele Jahre hinweg begleitet und mit vielen nützlichen Tipps ermutigt sowie Schritt für Schritt mich meinen anspruchsvollen Zielen nähergebracht haben.

Insbesondere bedanke ich mich auch bei meinem Betreuer, Herrn FH-Prof. Dipl.-Ing. Dieter Lutzmayr, der mir die Masterarbeit für die FH CAMPUS 02, auch durch die Zurverfügungstellung der notwendigen Ressourcen und Arbeitsmitteln ermöglicht hat, sowie bei den wissenschaftlichen Assistenten allen voran Herrn Dr. Ali Abdallah und Herrn Matthias Primas für Ihre wertvollen wissenschaftlichen Ratschläge und technischen Informationen sowie für die gute Zusammenarbeit.

KURZFASSUNG

Als Virtual Reality wird eine Computertechnologie von künstlich geschaffenen Welten, Produkten, Anwendungen, Fähigkeiten und Fertigkeiten bezeichnet. Sie ermöglicht dem Anwender sich in eine virtuelle Welt mit der Hilfe von Virtual Reality-Brillen zu teleportieren, in der er sich nahezu frei darin bewegen kann. Der Anwender nimmt durch die Virtual-Reality Brille seine reale Umgebung nicht mehr wahr und taucht in eine unglaublich real erscheinende immersive virtuelle Welt ein.

Die FH CAMPUS 02 plant ihr gesamtes Elektro-Energietechnik-Labor als virtuellen digitalen Zwilling abzubilden. Diese Abbildung soll dann das Gleiche ausführen, wie das Labor in der realen Welt dazu in der Lage ist. Das Ziel dieser Masterarbeit ist es, eine Schnittstelle über zwei Hochsprachen in C++ und C# zu entwickeln die es dem Anwender ermöglicht zwischen der realen und der virtuellen Welt wechselweise zu kommunizieren.

Das Ergebnis dieser Masterarbeit ist das Programmieren einer seriellen Schnittstelle die ein Arduino Board, mit einem angeschlossenen Sensor, der die Luftfeuchtigkeit und Temperatur zyklisch ermittelt und die Werte über einen Mikrocontroller speichert, mit Unity 3D 2018 sowie Modbus-TCP verbindet. In dieser Spiele-Engine sowie über Modbus-TCP werden die Ergebnisse in analoger und digitaler Form virtuell ausgegeben und visualisiert. Über eine einfache Schaltung werden, über ein selbst hergestelltes LED-Board für Demonstrations- und Testzwecke, die Abläufe dargestellt.

Die größte Herausforderung der Umsetzung dieser seriellen Schnittstelle stellt dabei dar, dass es dafür bisher keine wissenschaftlichen Ansätze gibt.

ABSTRACT

Virtual Reality symbolizes a computer technology which is capable of simulating artificial environments, products, applications, skills and abilities in a non-physical-world. Immersion into virtual reality constitutes the perception of being physically present utilizing sophisticated head-mounted-displays. Head-mounted-displays facilitate an immersive experience that seems more real than an experience provided by a desk-top-monitor.

The University of Applied Sciences CAMPUS 02 plans to map the entire Automation technology laboratory as a virtual digital twin. The aim of this master thesis is to develop a not yet existing interface in two different standard languages in C++ and in C#, mapping gauges in the virtual world, and creating thereby an interaction between the real and the virtual world for prospective applications in the laboratory.

For the design and the implementation of the interface transforming to an immersive virtual environment it is essential to code in the Arduino Integrated Development Environment and in the gaming engine Unity 3D 2018 through Visual Studio. An Arduino Board operates as a substitute for a real measurement device of the laboratory. This board is connected with a sensor and provides the possibility to measure the humidity and the temperature and displays the results through a microcontroller on a virtually adapted laboratory bench in Unity 3D 2018 and in a Modbus-TCP application. A newly developed prototype of a light-emitting diode board was utilised for demonstrating and testing purposes.

Up to the present there were no scientific papers for a comparable serial interface.

INHALTSVERZEICHNIS

1 Einleitung ... 1
2 Virtual Reality ... 3
2.1 Wesentliche Meilensteine in der Geschichte und der Beginn der kommerziellen Anwendung von Virtual Reality ... 4
2.2 Gartner Hype Cycle for Emerging Technologies ... 8
2.3 Zielgruppen für die Anwendung von Virtual Reality ... 9
2.4 Unterschied zwischen Augmented (AR) und Virtual Reality (VR) ... 10
2.5 Chancen und Risiken des Virtual Reality Booms ... 13
2.5.1 Chancen und Vorteile durch die Anwendung von Virtual Reality ... 14
2.5.2 Gefahren, Risiken und Nachteile bei Virtual Reality-Anwendungen ... 15
2.5.3 Motion-Sickness (Kinetose) ... 16
2.5.4 Simulator-Sickness ... 17
2.5.5 Sucht und Abhängigkeitspotential ... 17
3 Virtual Reality-Brillen (VR-Brillen) ... 20
3.1 Allgemeiner Aufbau der Virtual Reality-Brille ... 21
3.1.1 Kunststoffgehäuse ... 21
3.1.2 Display ... 22
3.1.3 Linsen ... 22
3.1.4 Gyroskop ... 24
3.1.5 Magnetometer ... 24
3.1.6 Beschleunigungssensoren ... 24
3.2 Unterteilung der Virtual Reality-Brillen gemäß ihrer Anwendung ... 24
3.3 Trackingsysteme bei Virtual Reality-Brillen ... 25
3.3.1 Headtracking ... 25
3.3.2 Positional-Tracking ... 26
3.4 Bildqualität der Virtual Reality-Brillen ... 26
3.5 Aktuelle massenmarkttaugliche VR-Brillen im Vergleich ... 27
3.6 Virtual Reality-Brillen im Highend-Bereich ... 31
3.6.1 Pimax 5K+ und 8K Virtual Reality-Brille ... 31
3.6.2 StarVR One (XT) Virtual Reality-Brille ... 33
3.6.3 5K Virtual Reality-Brille XTAL ... 34
3.7 Virtual Reality-Ganzkörperanzug (HoloSuit) ... 35
4 3D-Game-Software (3D Engines) ... 39
4.1 Spieleengine Unity 3D 2018 ... 39
4.2 Unreal Engine ... 40
5 Visualisierung und Virtuelle Steuerung eines Digital Twin-Modelles in Hard- und Software ... 42
5.1 Blackbox-Modellerstellung eines Digital Twins ... 43
5.2 Umsetzungslayout eines Digital Twin-Modelles ... 43
6 Arduino Nano-Board V3.0 ... 45

6.1 Arduino Hardwarekomponenten ... 45
6.2 Pinbelegung und Pinbeschreibung des Arduino Nano Boards ... 46
Anschlussmöglichkeiten, Bauteile bzw. Funktionen ... 49
7 Digitaler Temperatur- und Luftfeuchtigkeits-sensor DHT11 ... 52
7.1 Die Datenübertragung ... 53
7.2 Berechnungsbeispiel ... 54
8 Die LED-Leiterplatte (Protoyp eines LED-Boards) ... 55
8.1 Das Leiterplattendesign ... 55
8.2 Die Schaltungstechnik der Leuchtdioden (LEDs) ... 57
8.3 Bestimmen der Vorwiderstände (R_V) am LED-Board ... 57
8.4 Die Farbcodierungstabelle für bedrahtete Widerstände mit den 5 Farbringen ... 59
8.5 Definierte Regeln für das Leuchten der LEDs ... 60
8.6 Fehlerbehebung ... 61
9 Funktionen, Anwendung sowie Einrichtung der Virtual Reality-Hardware ... 62
9.1 Die HTC Vive Virtual Reality-Brille ... 62
9.2 Bedienung der HTC Vive Virtual Reality Motion-Controller (VR-Controller) ... 63
9.3 Die Einrichtung der zwei HTC Basis-Stationen ... 65
10 Softwareerstellung für die Interaktion mit der Virtuellen Realität ... 66
10.1 Arduino IDE Entwicklungsumgebung, Version 1.8.4 ... 66
10.2 Arduino-Softwareprogramm (sketch_nano) ... 68
10.2.1 Variablendeklaration ... 69
10.2.2 Konstanten und Präprozessor-Konstanten ... 69
10.2.3 Void setup () – void loop () ... 70
10.2.4 Schlüsselwörter ... 70
10.2.5 Switch-Anweisung ... 71
10.2.6 Eingangs-/Ausgangs-Ports ... 71
10.2.7 Schrittgeschwindigkeit (Baudrate) und Übertragungsgeschwindigkeit ... 71
10.2.8 Arduino-Library-Manager ... 71
10.2.9 Umsetzung des Programmcodes (sketch) ... 73
11 Spieleengine Unity 3D 2018 ... 76
11.1 Umgang mit und Steuerung von Unity 3D 2018 ... 76
11.1.1 Editor in Unity 3D 2018 ... 76
11.1.2 Steuerung in Unity 3D 2018 ... 79
11.2 Die Virtual Reality-Entwicklungswerkzeuge (SDK) ... 81
11.3 SteamVR (für die HTC Vive Virtual Reality-Brille) ... 82
11.4 Die Erstellung der einzelnen Skalen für das Hygrometer und das Thermometer und Einbindung in Unity 3D 2018 ... 83
11.5 Programmerstellung in Unity 3D 2018 ... 84
11.5.1 Softwarescript in Unity 3D 2018 ... 84
11.5.2 Verschiedene Datentypen in C# ... 88
11.6 Modellerstellung eines virtuellen Messtisches in Unity 3D 2018 ... 90
11.7 Trouble-Shooting in der Umsetzung in Unity 3D 2018 ... 91

11.8 Schnittstelle Modbus-TCP für Unity 3D 2018 93
12 Erkenntnisse, Ergebnisse und Ausblick 96
12.1 Erkenntnisse 96
12.2 Ergebnisse 96
12.3 Ausblick 97
Literaturverzeichnis 98
Abbildungsverzeichnis 101
Abkürzungsverzeichnis / Glossar 104

1 EINLEITUNG

Unser heutiges Leben und unsere moderne Arbeitswelt sind geprägt von Begriffen wie z.B. Industrie 4.0, Digitalisierung, *Machine-Learning*, Künstliche Intelligenz oder *Industrial Internet of Things* (IIoT)-Lösungen beispielsweise für den Einsatz von *smarten* Maschinen. Die „junge Generation", die sogenannten *Digital Natives*, wachsen mit diesen Begriffen unmittelbar auf und auch Personen aus der Generation davor, die im „Digitalen Zeitalter" leben und arbeiten, beschäftigen sich stark mit Dingen, wie der digitalen Abbildung von optimierten Geschäftsprozessen und der digitalen Transformation in Produktion und Logistik.

Bei all diesen Themen geht es vorwiegend um die Anbindung bestehender Geschäftsprozesse an die eigene IT-Landschaft. Diesem Trend begegnet man natürlich, oft mit Unterstützung von „Alexa und anderen Spracherkennungsprogrammen", auch im privaten Umfeld. Viele Haushalte haben bereits eine internetbasierte Raumsteuerung mit einer gekoppelten Sprachsteuerungs- und Spracherkennungssoftware im Einsatz, die auch über die Möglichkeit eines Fern- bzw. Remotezugriffes verfügt. Nicht nur in diesem Bereich kann man in den letzten Jahren einen enormen Fortschritt verzeichnen, da die Spracherkennungssoftware mittlerweile auch schon in der Lage ist, Zusammenhänge gut zu erkennen. Die IT-Landschaft hat sich in den letzten Jahren auch stark verändert und den Herausforderungen der Arbeitswelt angepasst. Schon längst arbeitet man höchst effektiv und effizient in der virtuellen Welt und teleportiert sich darin virtuell zu den gewünschten Örtlichkeiten, ohne sich in der realen Welt physisch bzw. geographisch weiterbewegen zu müssen. In der virtuellen Welt spricht man hier von *Locomotion*, die die Art und Weise dieser Art der Fortbewegung (z.B. das Teleportieren) beschreibt.

Das verfügbare Potential, das diese eigentlich gar nicht so junge Virtual Reality-Technologie bietet, ist allerdings noch lange nicht ausgeschöpft. Die enorme Weiterentwicklung im Bereich der Virtual Reality birgt Chancen und Risiken, kann aber auch dabei unterstützen die Stärken bzw. Vorteile der Anwender zu fördern und die Schwächen bzw. Nachteile zu reduzieren um sich als Leader in seinem Segment, sowohl im privaten als auch im beruflichen Umfeld zu etablieren und zu behaupten. Es ist in der heutigen Welt längst eine unumgängliche Anforderung geworden, mit neuen Trends sowie enormen Datenmengen (*Big Data*) zurechtzukommen und dem Zeitgeist ständig zu folgen um der oder die Klassenbeste in der eigenen Kategorie oder Branche zu sein bzw. auch weiterhin wettbewerbsfähig zu bleiben. Natürlich gibt es auch einige wenige, die diesem Trend nicht folgen und dennoch höchst erfolgreich sind. Eine der Hauptanwendungen von Virtual Reality und für die Virtual Reality-Brillen ist der *Gaming*-Bereich. Man verwendet besonders für die Programmierung hier *frameworks* und *engine*s, wie beispielsweise die Unreal Engine, die „WebVR mit A-Frame" oder Unity 3D. Wichtig ist hierbei, dass der Spieleentwickler das Virtual Reality-Design gut kennt. Bei der Entwicklung eines Virtual Reality-Games ist eine Virtual Reality-Brille nicht unbedingt notwendig, aber im Laufe der Programmierung sinnvoll, da das Spiel laufend getestet werden sollte (z.B. Testen, ob das Spiel flüssig läuft, oder Testen ob noch Design-Anpassungen notwendig sind, etc.). Die Spieleentwicklung für und der Markt rund um den Gamingbereich werden in dieser Masterarbeit allerdings nur sehr peripher gestreift und der Fokus sehr stark auf die Anwendungen, die *Features* und die Komponenten im industriellen- und Schulungsbereich gelegt.

Im theoretischen Teil dieser Masterarbeit sollen neben einigen ethischen und sozialen Einflussgrößen auch Auswirkungen des Interagierens mit der virtuellen Welt sowie die Abgrenzung zu Augmented Reality

analysiert werden. Es ist ein großer Vorteil und ein flexibler und verhältnismäßig kostentechnisch günstiger Faktor virtuell zu interagieren und seine Arbeit zu verrichten. Es gilt jedoch stets zu berücksichtigen, dass der Anwender natürlichen körperlichen Grenzen unterliegt. Um ein vollständiges Bild der Wechselwirkungen der Kopplung der virtuellen mit der realen Welt zu bekommen sollen daher neben den technischen Aspekten auch der Faktor Mensch in den Mittelpunkt gestellt werden um die physischen, sozialen und moralischen Auswirkungen des Virtual Reality-Booms zu untersuchen. Eine Analyse des aktuellen Standes der Technik sowie die verfügbaren Komponenten rund um die am Markt verfügbaren Virtual Reality-Brillen runden den theoretischen Teil der Masterarbeit ab und sollen einen Ausschnitt auf internationale Trends und zukünftige Entwicklungen zu Virtual Reality-Anwendungen geben.

Im praktischen Teil dieser Masterarbeit werden wesentliche technische Grundlagen, Ausprägungen und *Facetten* von Virtual Reality untersucht. Anhand eines virtuellen *Twins* eines realen Messtisches soll eine gekoppelte Anwendung der realen mit der virtuellen Welt analysiert werden. Es ist auch ein Ziel zu untersuchen, ob und auf welche Weise die Daten sowohl über die serielle Schnittstelle ein- und ausgelesen und über Modbus TCP über die virtuelle Umgebung ausgegeben werden können. Die Erweiterungs- und die Interaktionsmöglichkeiten zwischen der realen und der virtuellen Welt sind grenzenlos und es soll ein Ausblick dahingehend geschaffen werden, das Elektro-Energietechnik-Labor der FH CAMPUS 02 in der virtuellen Welt komplett abzubilden, sodass ein virtueller Zwilling (Digital *Twin*) entsteht. Die Vision ist es, dass die virtuelle Abbildung dann das gleiche ausführen soll, wie das Labor in der realen Welt dazu in der Lage ist. Die jeweilige Anwendung (z.B. der virtuelle Messtisch) soll angesteuert, visualisiert und kontrolliert werden.

Abb. 1: Virtual Reality Training,
Quelle: eLearning Industrie (2018), Online-Quelle [26.05.2019].

Eine weitere Zielsetzung beruht auf dem Ermitteln sowie der Analyse der vorhandenen Informationen bzw. Herausforderungen bei der Anbindung der realen an die virtuelle Welt, der vernetzten Anwendung unterschiedlicher Hochsprachen und der sprachenübergreifenden Programmentwicklung, dem Auslesen von Messwerten über ein Arduino kompatibles Nano-Board in der realen Welt, das mit einer LED-Board-Erweiterung als Prototyp dient, der virtuellen Visualisierung der Messergebnisse, dem Ermitteln von Rahmenbedingungen und Systemgrenzen sowie der gezielten Ansteuerung von Leuchtdioden über eine virtuelle Spieleumgebung in Unity 3D 2018 und der fließenden Interaktion mit der realen Welt.

Mittlerweile ist auch die Spiele-Engine Unity 3D 2019 am Markt verfügbar, jedoch ist diese für bereits bestehende Anwendungen nicht ausreichend abwärtskompatibel und daher konnte diese für die Erstellung des virtuellen Messtisches im Rahmen dieser Arbeit nicht eingesetzt werden.

2 VIRTUAL REALITY

Abb. 2: Global Virtual Reality Market,
Quelle: Reuters Editorial News (2018), Online-Quelle [30.04.2019].

Virtual Reality ermöglicht durch Eintauchen des Anwenders in eine „andere Realität“, vor allem durch Stimulation des visuellen und akustischen Sinns, eine konstruktionsgetreue Nachbildung von z.B. Maschinen und Anlagen in einer virtuellen Welt zu schaffen. Diese Nachbildungen inklusive ihrer Prozessmodelle und Funktionen können in der virtuellen Welt nicht nur simuliert werden, sondern es ist auch möglich interaktive Schnittstellen dieser in das reale Umfeld zu erzeugen. Mit Virtual Reality-Anwendungen können auch Trainings- und Weiterbildungsprozesse für die Bedienung und Handhabung von Maschinen und Anlagen sowie Werkzeugen virtuell durchgeführt werden. Transformationen des virtuell angeeigneten Wissens und der Kompetenzen, der zusätzlich erworbenen Fähigkeiten und Fertigkeiten aus der virtuellen Lernumgebung können anschließend einfach und effizient in der realen Welt angewendet und umgesetzt werden. Craftguide, ist beispielsweise ein 2018 gegründetes *Start-up*-Unternehmen und Anbieter von Virtual Reality- und Augmented Reality-Schulungen sowohl in Handwerk als auch in der Industrie. Craftguide hat es sich konkret zum Ziel gesetzt, handwerkliche Wissensvermittlung des Bau-, Holz-, Elektro- und Metallgewerbes durch Virtual Reality-Brillen für jedermann und überall zugänglich zu machen. Der Lernprozess spricht unterschiedliche Lerntypen (visuell) und Lernbereiche (kognitiv und psychomotorisch) an. Die Schulungen in der virtuellen Umgebung, z.B. ein Virtual Reality-Wartungsszenario, sollen zum Autodidakten-Lernen von handwerklichen Arbeitstechniken, dem fachgerechten Umgang mit auf original-CAD-Daten basierten Maschinen und der Verbesserung der Arbeitssicherheit durch gefahrlose Maschinen- und Sicherheitseinweisungen führen. Technologien und Verfahren werden dafür in die virtuelle Umgebung transferiert indem die 3D-Daten digital aufbereitet, mit handwerklichem Fachwissen kombiniert und visualisiert werden. Aktuell stehen dafür die Virtual Reality- und Augmented Reality-Anwendungen über HTC Vive und Microsoft HoloLens (einer Mixed Reality-Brille) zur Verfügung.[1]

Man bewegt sich zwischen den Welten und über die Systemgrenzen hinweg und bringt die reale in die virtuelle Welt und interagiert in dieser wiederum mit der realen Welt. Das birgt hohe Anforderungen an die Modellierung und Konzepterstellung, an die Softwareprogrammierung und an technische Systemlösungen in sich und ist durchaus in der Interaktion zwischen den Schnittstellen und im Ausmaß, wie Virtual Reality das Leben der Menschen zu beeinflussen vermag, auch sehr konfliktbehaftet. Gemäß den Visionen verschiedener, voneinander unabhängiger Hersteller von Virtual Reality-Equipment soll Virtual Reality schon in 2 bis 3 Jahren in jedermanns Haushalt im Einsatz sein. Diese Vision wird von der Tatsache unterstützt, dass die Virtual Reality-Technologie von Tag zu Tag weiterentwickelter, leistungsfähiger, leichter zugänglich, kostengünstiger und mit anderen Geräten weitaus kompatibler ist.

[1] Vgl. Virtual Reality Magazin (2019), Online-Quelle [27.05.2019].

2.1 Wesentliche Meilensteine in der Geschichte und der Beginn der kommerziellen Anwendung von Virtual Reality

Die folgenden Tabellen und Abbildungen, die aus zahlreichen Internetquellen zusammengefasst sind, zeigen die Geschichte von Virtual Reality zwischen 1838 bis 2014. Es soll damit unterstrichen werden, wie viel Aufwand und Zeit investiert werden musste, um diese aufwendige Technologie zu dem zu machen, was sie heute ist. Dabei stehen wir auch heute immer noch am Anfang deren kommerzieller Nutzung.

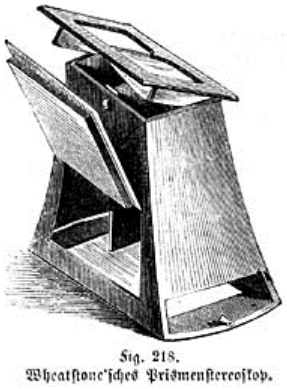
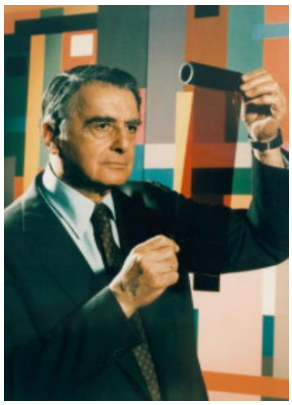
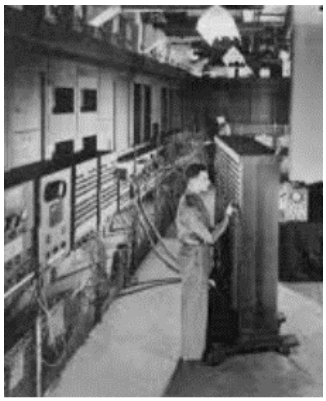
 Fig. 218. Wheatstone'sches Prismenstereoskop.		
Sir Charles Wheatstone erfand mit dem ***Stereoscope*** ein Gerät, das auf binokularen Bildern basierte und somit die Illusion eines 3-dimensionalen Objektes erstellen konnte. In vielen europäischen Städten entstanden im ausgehenden 19. Jahrhundert sogenannte Kaiser-Panoramen. Darunter versteht man zylindrische Schaukästen, mit deren Hilfe einige Personen gleichzeitig stereoskopische Bilder betrachten konnten. Im Inneren wurden Dia-Aufnahmen im Kreis um jeweils einen Zuschauerplatz weiterbewegt. Erst der Kinematograf konnte die Kaiserpanoramen verdrängen.	Der amerikanische Physiker Edwin Herbert Land konstruierte 1932 einen Polarisationsfilter (**Polarisator-Brille**) aus makromolekularem Kunststoff, dem sogenannten Polaroid-Filter. Im Jahr 1933 wurde schließlich das Patent darauf erteilt. Noch heute werden in 3D-Filmen Polarisationsfilter eingesetzt, um die beiden von zwei verschieden Punkten aufgenommenen, übereinander projizierten Bilder dem rechten bzw. linken Augen zuzuführen.	Der Computerpionier John Presper Eckert (Hardwareentwicklung) und der amerikanische Physiker sowie Computer Ingenieur John William Mauchly (konzeptuelles Design) erfanden gemeinsam die erste elektrisch-numerische Rechenwerk- und Universalrechenmaschine *Electrical Numerical Integrator and Calculator* (**ENIAC**). Diese gilt als Vorläufer des modernen Heimcomputers. Für ENIAC wurden 17.468 Vakuumröhren, 70.000 Widerstände, 10.000 Kondensatoren und 1.500 Relais verbaut. Dieser benötigt eine Stellfläche von 167 m² und wiegt 30.000 kg. ENIAC wurde 1946 für ballistische Tabellen militärisch eingesetzt.
1838 bis 1851	1932	1946

Tabelle 1: Meilensteine in der Entwicklung von Virtual Reality von 1830 bis 1950, Quelle: Eigene Darstellung.

Die Meilensteine in der Entwicklung von Virtual Reality von 1950 bis 1970 wie z.B. der Entwicklung des Sensorama's und des Damokles Schwertes konnten nicht innovativ umgesetzt werden.

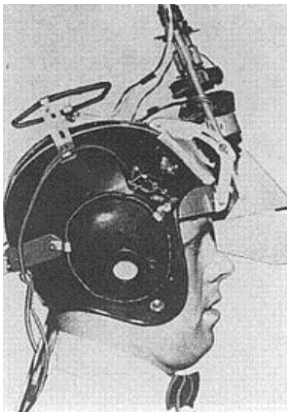
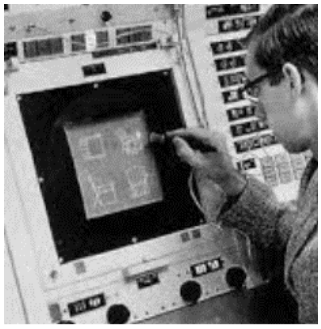
			
1956 entwickelte und 1962 patentierte Morton Heilig das **Sensorama**. Es war eines der ersten bekannten Beispiele von immersiver, multisensorischer Technologie, das in der Lage war 3D-Bilder mit einem breiten Sichtfeld abzuspielen. Es diente dazu, Filme realitätsnahe zu erleben. Stereolautsprecher, ein stereoskopisches 3D-Display, Wind- und Geruchsgeneratoren sowie eine eingebaute Rüttelmechanik stimulierten die menschlichen Sinne. Das Sensorama blieb aber ein Einzelstück.	Zwei Ingenieure von Philco Corporation konstruierten „***Headsight"*** für militärische Zwecke. Das *Headsight*-Kopfteil bot je einen Bildschirm für das jeweilige Auge und verfügte bereits über magnetisches *Motiontracking*. Gekoppelt mit ferngesteuerten Kamerasystemen konnte jede Kopfbewegung des *Headsight*-Trägers auf die Kameraobjektive übertragen werden. So konnten Anwender damit sehr gefährliche Umgebungen aus sicherer Entfernung überwachen.	Der Wissenschaftler Ivan Edward Sutherland entwickelte ab 1962 am Massachusetts Institute of Technology (MIT) das Programm „***Sketchpad"***, welches als eines der ersten interaktiven Grafikanwendungen angesehen wird. Die Anwendung lief auf einem dafür umgebauten Lincoln-TX-2-Rechner und nutzte bereits einen Röhrenbildschirm. Durch dieses wurde der Umgang mit den Computern revolutioniert. Sutherland legte damit auch den Grundstein für die grafische Nutzung des *Desktops*.	**Damokles Schwert**: Ivan Sutherland und der Student Bob Sproul entwickelten das erste richtige Virtual Reality-Headset (*HMD*), das direkt an einen Computer angeschlossen wurde. Die erzeugten Bilder wurden auf einem augennahen Bildschirm dargestellt, um auf diese Weise das Fenster in eine virtuelle Welt zu suggerieren. Auf Grund seiner Größe und des Gewichtes benötigte es eine Aufhängung an der Decke, da es vom Anwender nicht getragen werden konnte.
1956 und 1962	1961	1962 bis 1963	1965 bis 1968

Tabelle 2: Meilensteine in der Entwicklung von Virtual Reality von 1830 bis 1950, Quelle: Eigene Darstellung.

Nach den ersten Durchbrüchen bezüglich Virtual Reality (VR) blieb es lange Zeit still um dieses Thema, da der Computer noch nicht leistungsstark genug und die VR-Technologie noch nicht ausgereift waren.[2]

[2] Vgl. VR-NERDS, A Virtual Reality Showcase (2017), Online-Quelle [12.08.2019].

Es wird angenommen, dass 1982 der Begriff „Virtual Reality“ zum ersten Mal im Roman „*The Judas Mandala*“ von Damien Francis Broderich angeführt wurde. Bereits im Jahr 1984 wurde Virtual Reality schließlich durch den wegweisenden Roman „Cybermancer“ einem breiten Publikum bekannt gemacht.

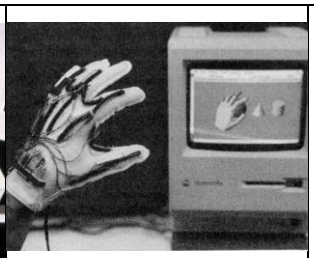
			
Die NASA entwickelte 1985 das *Virtual Environment Display Headset (***VIVED)**, welches Astronauten und Kosmonauten helfen sollte, ihre Roboter aus der Entfernung sicher zu kontrollieren, sodass sie riskante Operationen außerhalb des Raumfahrzeuges nicht selber durchführen mussten. Das HMD hat eine Auflösung von 320 x 240 Pixel. Im VPE-Projekt (*Virtual Planetary Exploration*) versuchte die NASA die enormen Datenmengen, die die Viking Sonde vom Mars funkten, zu visualisieren.	Thomas Zimmermann und Jaron Lanier gründeten die VPL (*Visual Programming Language*) und entwickelten den ***Data Glove (Hand Gesture Interface Device)***. Dieser war an der Handoberseite mit Glasfasern bestückt um Fingerdaten zu erfassen. Es war nun erstmals möglich, ein Körperteil zu digitalisieren und für Interaktionszwecke in eine virtuelle Welt zu übertragen. Der Datenhandschuh war der erste kommerzielle Datenhandschuh und kostete rund 9.000 US-$. 1993 ging VPL in Konkurs.	1992 wurde *Cave Automatic Virtual Environment* (**CAVE**) auf der Siggraph vorgestellt. Entwickelt wurde es vom Kunstprofessor Daniel Sandin, vom Informatiker Tom DeFanti sowie von Carolina Cruz Neira an der University of Illinois at Chicago. CAVE ist ein Virtual Reality-Visualisierungssystem mit bis zu 6 festen, senkrecht zueinanderstehenden Projektionsebenen. Auf diese Flächen werden 3D-Bilder geworfen, sodass der Anwender tatsächlich inmitten der auf dieser Weise virtuell erzeugten Umgebung steht.	Nintendo stieg in den VR-Markt ein und brachte den ***Virtual Boy*** auf den Markt. Dieser ähnelte einer Tauchmaske auf einem Gestell und wurde mit einem zusätzlich angeschlossenen Controller bedient. Er muss auf einer ebenen Fläche stehen und wie ein fest installiertes Fernglas genutzt werden. Durch leicht versetzte Bilder entsteht ein sanfter 3D Effekt. Das Bild wird durch rote Leuchtdioden auf schwarzem Grund dargestellt, wodurch das Bild monochrom ist. Die Auflösung der erzeugten Bilder beträgt 384 x 224 Pixel.
1985	1982 bis 1989	1992	1995

Tabelle 3: Meilensteine in der Entwicklung von Virtual Reality von 1980 bis 1999, Quelle: Eigene Darstellung.

In den 90er Jahren war die Virtual Reality-Technik für die breite Masse noch nicht bezahlbar, denn diese kostete zu dieser Zeit noch zwischen 50.000 bis 100.000 US-$.

Das erste Virtual Reality-Konzept reichte bis ins 19. Jahrhundert zurück und zeigte, dass die Technologie rund 180 Jahre gebraucht hat um sich zu entwickeln um ab dem Jahr 2014 einen richtigen Aufschwung zu erleben.

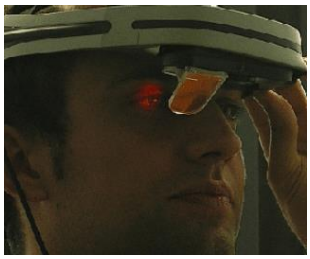
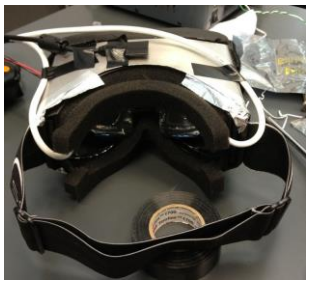
			
Die amerikanische Firma **Microvision** Inc. entwickelte im Jahr 2000 für die US-Airforce ein Virtual Reality Head-Mounted-Display (HMD), das erstmals eine Auflösung von 1920 x 1080 Pixel erreichte. Über den Preis und das Gewicht wurde bis heute Stillschweigen bewahrt.	2009 erforschte das Frauenhofer Institut IPMS eine interaktive **Datenbrille mit *Eye Tracking***. Mit dessen Hilfe können Anwendungen durch Augenbewegungen gesteuert werden. Sie zeigt bidirektional Informationen an und nimmt auch Befehle entgegen. Im gleichen Jahr wurde am *Ars Electronica Center* in Linz der Nachfolger des CAVE mit der Virtual Reality-Großrauminstallation **DEEP SPACE** eingeweiht. Sie stand Künstlern für ihre Arbeiten zur Verfügung. Diese basiert auf einer alten Technologie mit 16 x 9 m großen, ultrahochauflösenden Bildern.	Der amerikanische Spieleentwickler und Chef-Programmierer John Carmack von id-Software stellte eine selbstgebastelte leistbare Virtual Reality-Brille vor, die bei einem Komponentenpreis von rund 500 US-$ die Performance von Profi HMDs erreichen sollte, die sonst 10.000 US-$ oder mehr kosteten. Die Brille wies mit 90 x 110 ° ein spielegeeignetes großes *Field of view* auf und arbeitet mit einer Latenzzeit von lediglich 20 ms. Durch die kurze Latenz merkte der Spieler keine Verzögerungen im Bildaufbau. Im gleichen Jahr gründete Palmer Luckey die Firma **Oculus**.	**Hype:** 2014 entpuppte sich als das Jahr, in dem Virtual Reality einen richtigen Aufschwung erlebte. Facebook kaufte die Firma Oculus für ca. 2,3 Mrd US-$. Die Oculus Rift DK2 Reactions wurde ausgeliefert. Auf der US-Game Developers Conference (GDC) in San Francisco stellte Sony den Prototypen seines HMD für die PS4 vor. Ende des Jahres stellte Oculus auf der Entwicklungskonferenz „Oculus Connect“ den neuen Prototypen „*Cresent Bay*“ vor, der technische bereits sehr nah an der für 2015 erwarteten *Consumer* Version der Oculus Rift lag.
2000	2009	2012	2014

Tabelle 4: Meilensteine in der Entwicklung von Virtual Reality von 2000 bis 2014, Quelle: Eigene Darstellung.

Im Jahr 2019 ist durch fehlende „Killer“-Applikationen das Interesse an der Virtual Reality-Technologie bei der breiten Masse wieder ein wenig abgeflaut. Dafür gibt es bereits eine große Menge an effizienten Spezialanwendungen und komplexe technologische Herausforderungen im professionellen beruflichen Umfeld der Anwender.

Besonders hervorzuheben ist es, dass die Virtual Reality-Systeme kontinuierlich leistungsfähiger, preiswerter und die Komponenten zusätzlich immer kleiner werden. Zudem werden die Entwicklungsframeworks und Schnittstellen auch stetig weiterentwickelt und somit ausgereifter.

2.2 Gartner Hype Cycle for Emerging Technologies

Eine wichtige Orientierungshilfe für den technologischen Lebenszyklus von Virtual Reality bietet der jährlich aktualisierte Hype Cycle für neue Technologien von Gartner (siehe dazu auch Abb. 3). Der Hype Cycle stellt dabei dar, wann eine Technologie innovative Ansätze mit sich bringt.

Auf der Y-Achse wird der Grad der erkannten Aufmerksamkeit des Marktes und von Investoren und auf der X-Achse die vergangene Zeit seit Bekanntwerden der Technologie aufgetragen, wobei jeder Zyklus in fünf Entwicklungsphasen unterteilt wird, die etwa 2 bis 5 Jahre dauern können.

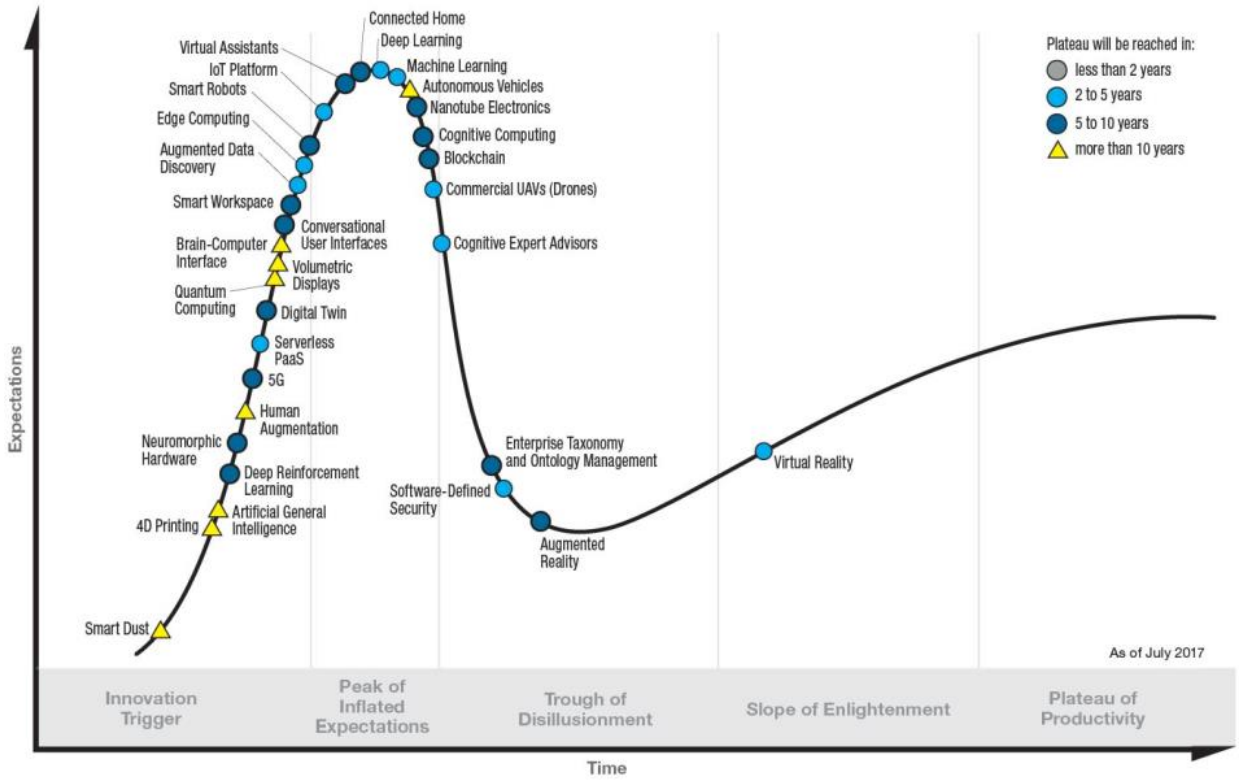


Abb. 3: Gartner Hype Cycle for Emerging Technologies, Quelle: Gartner.com (2017), Online-Quelle [24.11.2019].

Gartner’s Hype Cycle vom Juli 2017 zeigt gemäß mehrerer Internetquellen, dass die Virtual Reality-Technologie das „Tal der Enttäuschungen“ (die Technologie konnte die zuvor aufgebauten teilweise utopischen Erwartungen nicht erfüllen, und daher sinkt die Aufmerksamkeit an dieser Technologie und es folgt die Desillusionierung) hinter sich gelassen hat und sich aktuell auf dem „Pfad der Erleuchtung“ (es entsteht ein realitätsnahes Bewusstsein und die Technologie wird hinsichtlich ihrer Stärken und Schwächen und welche Grenzen und welchen Nutzen dies Technologie hat betrachtet) befindet und den Prognosen

von Gartner zufolge in den nächsten 2 bis 5 Jahren zum „Plateau der Produktivität“ (die neuen Möglichkeiten durch diese Technologie werden als Vorteile erkannt und Geschäftsmodelle werden entwickelt) aufsteigen wird.

Am Ende des Gartner Hype Cycle steht für Technologien und Geschäftsmodelle die Phase, in der sie schließlich massentauglich werden. Marktreife Produkte bewähren sich unter rationalen Erwartungen. Spätestens hier rücken die Technologien wieder zurück in den Blick der Investoren, die sie im „Tal der Enttäuschungen“ zunächst aufgegeben hatten.

2.3 Zielgruppen für die Anwendung von Virtual Reality

Im Folgenden sollen daher die typischen Zielgruppen, auf die die Anwendung von Virtual Reality besonders zu fokussieren ist, auch eingehend analysiert werden. Am besten bietet sich dafür eine nähere Untersuchung der unterschiedlichen Generationen, wie beispielsweise der „*Generation Y*“ und der „*Generation Z*“ an.

Die nachfolgende Abbildung zeigt eine Übersicht über die verschiedenen Generationen, um in weiterer Folge eine Analyse der typischen Ausprägungen und Merkmale der jeweiligen Generation durchzuführen.

Abb. 4: Überblick über die Generationen, Quelle: Absolventa GmbH (2019), Online-Quelle [26.05.2019].

Die Absolventa GmbH, als Expertin für das *Recruiting* von Nachwuchstalenten und Tochtergesellschaft der FUNKE Mediengruppe, kategorisiert die Generationen wie folgt:[3]

Die „Traditionalisten“ wurden zwischen 1922 und 1945 geboren und die Babyboomer zwischen 1946 und 1964. Letztere sind die erste Nachkriegsgeneration und zählen zur geburtenreichsten aller Generationen. Sie sind vor allem dadurch durch das Schaffen von Strukturen und dem Wunsch nach Entschleunigung gekennzeichnet.

Die „*Generation X*“, sie werden auch „Generation Golf“ bzw. „*Digital Immigrants“* genannt, werden stark durch die Wirtschaftskrise geprägt. Diese „Gen X“ ist kennzeichnet durch Merkmale wie „Arbeit ist Mittel zum Zweck“, Unabhängigkeit und Technik. *Digital Immigrants*, müssen erst lernen mögliche innere Widerstände und Konflikte, bei der Verwendung von Virtual Reality-Brillen um in die virtuelle Welt einzutauchen, abzubauen, diese sind z.B.:

- die Angst vor Manipulationen,
- die Angst davor ein gläserner Mensch zu werden,
- die Angst vor Kontrollverlust über den eigenen menschlichen Körper
- die Angst vor Entscheidungsverlust und

[3] Vgl. Absolventa GmbH (2019), Online-Quelle [27.05.2019].

- die Angst vor „Gefahren und Einflüssen von außen“, die auf den schutzlosen Körper während man sich in der virtuellen Welt befindet, einwirken können

um somit zu verinnerlichen, dass mit dieser weiterentwickelten Technologie grundsätzlich nichts Neues kommt, sondern man diese Weiterentwicklung einer bereits bekannten Technologie positiv nutzen kann um Aktivitäten und Tätigkeiten schneller, ressourcenschonender und effektiver durchzuführen. Viele Anwender dieser Generation haben auch Angst vor Entwicklungen, die sie vermeintlich nicht mehr steuern, beeinflussen oder vorhersehen können. Auch diese Generation zählt grundsätzlich nicht zur primären Zielgruppe für die Anwendung von Virtual Reality.

Bei der „*Generation Y*“ sieht es nun in Bezug auf Anwendung und Akzeptanz von Virtual Reality schon ganz anders aus. Diese Generation wird auch die „*Millennials“* genannt. Die „Gen Y“ ist vor allem durch den Internetboom und die Digitalisierung geprägt und sie endet mit dem Auftreten der ersten *Digital Natives*. Ihre wichtigsten Ziele sind die Ausrichtung auf Projektarbeit und das Schaffen von Freiräumen für Privates. Sie haben den Begriff *Work-Life-Balance* und *Work-Life-Blend* (das bedeutet, Privates soll auch während der Arbeitszeit erledigt werden können) geprägt und besitzen zumeist ein hohes Bildungsniveau. Die Deloitte Studie „*Millennial-Survey* 2019“ attestiert der Gen Y eine pessimistische Einschätzung ihrer weiteren Zukunft und die Sehnsucht nach bildschirmfreier Zeit. Der Begriff *Digital Detox*, darunter versteht man eine Auszeit von der Online-Welt, ist zweifellos damit verbunden. Die Studienverfasser bezeichnen sie als die „Generation der Zerrissenen“ und als „Generation der vielen Optionen“.

Für die „*Generation Z*“ ab 1994, sie wird auch „*Digital Natives“* bzw. „*Generation YouTube*“ genannt, ist Digitalisierung Teil des Alltags. Das Ziel der „Gen Z“ ist die Selbstverwirklichung und es gibt keine Abgrenzung zwischen virtuell und real mehr.

2.4 Unterschied zwischen Augmented (AR) und Virtual Reality (VR)

Die Begriffe *Virtual Reality* („Virtuelle Realität“ bzw. „Virtuelle Welt“) und *Augmented Reality* („Erweiterte Realität“) werden im normalen Sprachgebrauch oft missbräuchlich bzw. falsch eingesetzt. Darunter versteht man grundsätzlich nämlich zwei Technologien mit einer komplett unterschiedlichen Zielsetzung in einem unterschiedlichen oder ähnlichen Arbeitsumfeld.

In der *Mixed Reality* werden beide Technologien schließlich miteinander verbunden bzw. vermischt und somit die virtuelle und die reale Welt miteinander gekoppelt.

Die wesentlichen Unterschiede zwischen Augmented und Virtual Reality können (wie auch in der Abb. 5 und 6 dargestellt) wie folgt definiert werden:[4]

- *Virtual Reality* (VR)-Technologien ermöglichen dem Anwender das Interagieren in einer Welt, die vollständig virtuell, also künstlich geschaffen wurde und in die auf eine realistische Art und Weise eingetaucht werden kann. Das reale Umfeld wird dabei vom Anwender nicht mehr wahrgenommen, da es komplett ausgeblendet wird. Der Einsatz der virtuellen Realität z.B. für die Durchführung von Simulationen zur Bedienung von Maschinen, der Assemblierung von Anlagenteilen, dem

[4] Vgl. Korgel (2018), S. 11.

Membrantausch in Membranventilen oder der richtigen Durchführung von Messungen an Messgeräten ist nicht neu, jedoch hat ein Wandel der Technologie in Bezug auf Kosten- und Dimensionsfaktoren zu einem großflächigeren Einsatz im beruflichen und privaten Umfeld geführt. Durch die Erweiterung z.B. mit realen Bildern wird Virtual Reality zur Augmented Reality („Erweiterte Realität").

- *Augmented Reality* (AR) oder „Erweiterte Realität" bedeutet die Erweiterung der „Realen Welt" mit virtuellen Informationen und der zusätzlichen Darstellung, Einbindung oder Überlagerung mit Daten (z.B. durch das Einblenden von Texten, siehe dazu auch Abb. 5) oder anderen Elementen (z.B. durch das Einblenden von Bildern und Videos) direkt in das Sichtfeld des Anwenders. Im Gegensatz zur „Virtuellen Welt", die vollständig computergeneriert ist, stellt Augmented Reality also eine mit Zusatzinformationen angereicherter bzw. erweiterte Wirklichkeit (Realität) dar.
  Diese Darstellung kann entweder über Virtual Reality-Brillen, einem Smartphone, welches das Kamerabild augmentiert, oder auch über ein Hologramm erzeugt werden. Im Unterschied zu Virtual Reality kann der Anwender bei Augmented Reality die Realität um ihn herum wahrnehmen, da diese nicht ausgeblendet wird. [5]

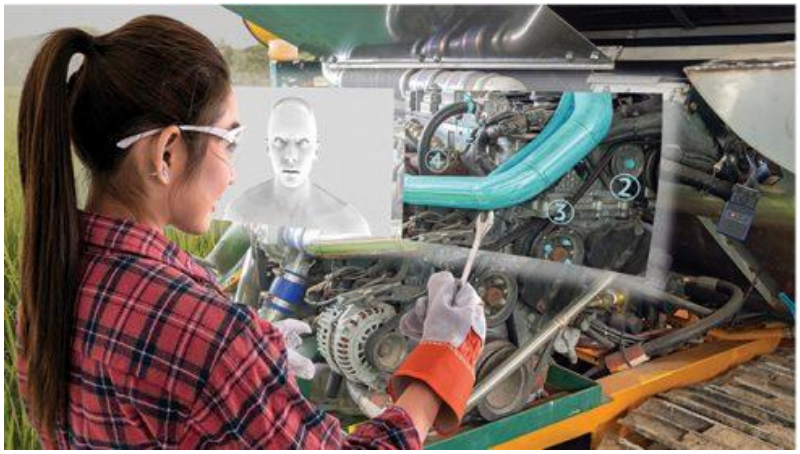


Abb. 5: The Reality of Augmented Reality, Quelle: SD Times (2018), Online-Quelle [30.08.2019].

1968 entwickelte Ivan Sutherland das erste funktionsfähige *Head Mounted Display* (HMD), das als Vorgänger der heutigen Augmented Reality Technologie eingestuft werden kann. Es erlaubt mit einem semi-transparenten Display die Realität um virtuelle Objekte zu erweitern, deren Lage mittels Kopfbewegungen verändert werden kann.

Die Anwendung der Augmented Reality hat seinen eigentlichen Ursprung in der Militärtechnik. Jetpiloten bekommen die wichtigsten Flugparameter wie beispielsweise die Geschwindigkeit, Flughöhe oder Fluglage in den Sichtbereich der Cockpitscheibe eingeblendet, um sich voll auf den Flug konzentrieren zu können. So haben die Piloten stets den vollen Überblick über sämtliche wichtige Flugparameter.

Nach demselben Prinzip arbeiten auch die Datenbrillen. Durch diese wird die „Reale Welt" betrachtet und zusätzlich werden im Sichtbereich passende Informationen zur Umgebung in einem

[5] Vgl. Thomas/Metzger/Niegemann/Welk/Becker (2018), Teil I, S. 4 ff.

halbtransparenten Monitor eingeblendet. Diese Informationen können geographische Angaben, technische Daten oder auch beispielsweise aktuelle Umweltinformationen sein. Erste kommerzielle Anwendungsfelder liegen in der Logistik und im Lagermanagement, bei der den Anwendern alle wichtigen Daten zu den eingelagerten Artikeln und Paketen im Sichtfeld angezeigt werden.[6]

- *Mixed Reality* bedeutet die Vermischung der realen Umgebung („Reale Welt") mit der virtuellen Realität („Virtuelle Welt"). Die „Reale Welt" und die „Virtuelle Welt" stellen die jeweiligen System-Grenzen der zwei Realitäten dar. Aber so wie die „Reale Welt" mit virtuellen Informationen bzw. Inhalten angereichert werden kann („Erweiterte Realität"), kann auch die „Virtuelle Welt" mit realen Informationen bzw. realen Einflüssen erweitert werden („Erweiterte Virtualität"). [7]

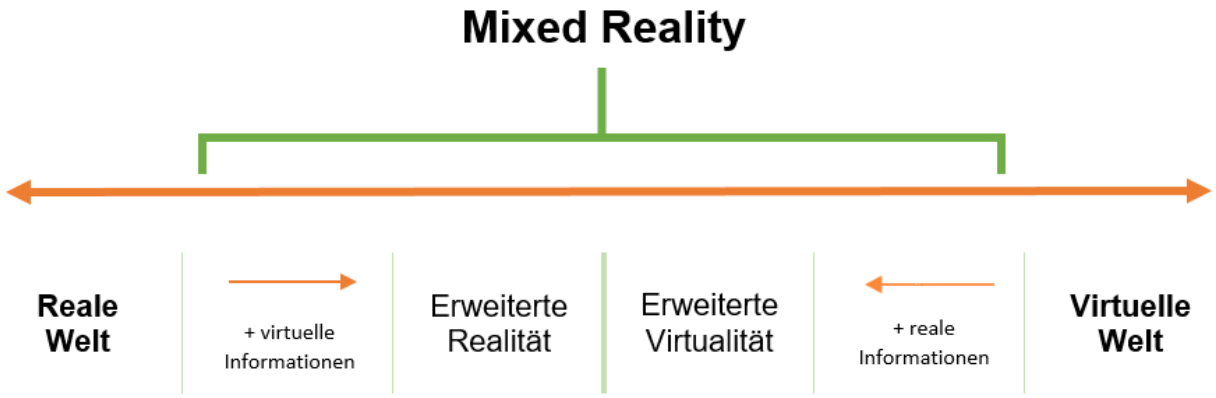


Abb. 6: Unterschied zwischen Augmented und Virtual Reality, Quelle: Eigene Darstellung.

„Die Erzeugung von Reizen ist hier auch nur eine Aufgabe auf dem Weg zur perfekten virtuellen Realität. Der Mensch möchte die Welt nicht nur betrachten und fühlen, sondern auch in dieser Welt handeln. Dies erfordert, dass die „Virtuelle Welt" simuliert wird, die Aktionen des Menschen der Simulation bekannt sind und diese Aktionen so die Simulation beeinflussen können. Die Ergebnisse der Simulation haben wiederrum Auswirkungen auf die Erzeugung von Reizen, denn bewegt sich der Mensch in der Virtuellen Realität, muss auch die Reizerzeugung die neue Position berücksichtigten. In der „Virtuellen Welt" ist man allerdings auch nicht zwangsläufig an die Gesetze der „Realen Welt" gebunden. Für die „Virtuelle Welt" müssen eine Vielzahl von Bildern pro Sekunde erzeugt werden, damit der menschliche Anwender Bewegungen in der „Virtuellen Welt" als kontinuierlich und natürlich wahrnimmt. Die benötigte Anzahl an Bildern pro Sekunde ist abhängig vom Anwender und seiner aktuellen Konstitution (ob es sich um einen nüchternen oder einen betrunkenen Anwender handelt). Typischerweise werden 60 Bilder pro Sekunde für die Erfüllung der Forderung nach Echtzeit benötigt (*frames per second*). Bei der Erzeugung von haptischen Reizen gilt es noch größere Echtzeitbedingungen zu erfüllen. In der Regel müssen Informationen mindestens 1.000-mal pro Sekunde von der Grafikkarte an das jeweilige Ausgabemedium gesendet werden."[8]

[6] Vgl. Virtual Reality bei Conrad (2017), Online-Quelle [28.08.2019].

[7] Vgl. Schröter (2012), Nr. 51, S. 104 – 120.

[8] Dörner (2013), S. 5 – 7.

Einer der Vorteile der Augmented Reality- und Virtual Reality-Technologien liegt daher auch in der zeitlichen und räumlichen Flexibilität der Anwender. Zur Identifikation von Fehlern an Maschinen und die anschließende Reparatur durch Augmented Reality-basierte Anleitungen bringt gemäß dem Frauenhofer-Institut für Entwurfstechnik Mechatronik IEM zusammen mit dem Frauenhofer-Institut für industrielle IOSB-INA neue Anwendungen wie die *Remote Maintenance Assistance*, Immersives Training und smarte Assistenzsysteme hervor, um beispielsweise Ausfallszeiten zu verkürzen, Reisekosten zu reduzieren und Produktionsausfälle zu reduzieren.

2.5 Chancen und Risiken des Virtual Reality Booms

Im Zeitalter der Industrie 4.0, dem „Digitalen Zeitalter“, ist der selbstoptimierte und selbstmotivierte „Maschinenmensch“ bzw. der *Digital Native* ohne Sozialbeziehungen am Arbeitsmarkt sehr gefragt und soll jederzeit für die Arbeitgeber verfügbar sein. Solange er im Arbeitsleben steht soll er idealerweise immer gesund und fit bleiben und weder durch Familie, Freunde oder persönliche Interessen eingeschränkt sein.

Laut den Autoren der „Fast Forward 2030“ Studie sind die *Digital Natives* fähig, große Netzwerke zu managen, viele Informationen aufzunehmen und effizient zu filtern. Sie arbeiten mit „Künstlicher Intelligenz“ und maschinellem- bzw. virtuellem Lernen. Aber sie haben kaum noch die Fähigkeit zur effizienten zwischenmenschlichen Kommunikation entwickelt und es wird, so die Aussage der Studie, 2030 wieder notwendig sein, diese Generation aus der Isolation herauszuholen, die durch die vielen *Devices* und Interaktionen in der digitalen und virtuellen Welt entsteht. In der „*New Work*“ bzw. am „*Future-Workplace*“ sind Begriffe wie *Work-Life-Balance*, Freiheit, zeitlich ungebundenes Arbeiten und Ortsungebundenheit nicht mehr wegzudenken.

Um als moderner Arbeitnehmer noch flexibler, dynamischer und kostenoptimierter für den Auftraggeber zu sein, werden *Remotejobs* zusehends interessanter. 95,6 % der Arbeitnehmer, die einer Bürotätigkeit nachgehen, verrichten gemäß einer 2018 von Citrix Systems, Inc., einem US-amerikanischem Software-Unternehmen, beauftragten Studie in Österreich diese immer noch vom Büro aus.

Hier greifen die Chancen und Vorteile des Arbeitens mit virtuellen Anwendungen voll durch. Berufliche Anforderungen können simuliert und in der virtuellen Welt visualisiert werden. Ergebnisse können dadurch auch rasch einem breiten globalen Publikum präsentiert und vorgestellt werden, ohne dass die Teilnehmer dafür an einem Tisch zusammensitzen müssen. Das spart Ressourcen wie Zeit und Geld und bringt Flexibilität, aber beinhaltet auch Unverbindlichkeit. Mit der Modellierung von ultrarealistischen Avataren ist es für den Anwender sogar problemlos möglich, die Gesten, die Mimik und die Emotionen der Teilnehmer über die virtuelle Welt aufzunehmen.

EVR Studio Co Ltd, ein 2016 gegründetes und in Seoul (Korea) ansässiges Unternehmen mit rund 50 Mitarbeitern, hat sich beispielsweise auf fotorealistische Avatare spezialisiert. Unter einem Avatar versteht man eine virtuell erstellte Figur bzw. Person, die einem Anwender in der virtuellen Welt zugeordnet wird. Sie dient als optische Identifikationsfigur (siehe dazu auch die nachfolgende Abbildung, die in der Unreal Engine erstellt und von EVR Studio präsentiert wurde), zu denen die Anwender in der virtuellen Welt durchaus auch eine emotionale Bindung aufbauen können. EVR Studio konzentriert sich derzeit vorwiegend auf die Entwicklung von Virtual Reality-Spielen, mobile VR-Streaming-Lösungen und VR-Videos über das interne EVR LAB.

Abb. 7: Das EVR Studio VR Projekt *„Project M"-Teaser*, Quelle: MIXED (2017), Quelle: Eigene Darstellung.

Der US-Philosoph John Searle meint im übertragenen Sinn, aus dem Themenbereich der „Künstlichen Intelligenz" abgeleitet dazu, dass man weit davon entfernt sei, Computer und Avatare mit Bewusstsein konstruieren zu können, einfach, weil man nicht wisse, wie Bewusstsein entstehe. Ein Computer und damit ein Softwareprogramm basieren nur auf einer Syntax und keiner Semantik. Es fehle ihnen jeglicher Sinn für Bedeutung und das ist wohl ein gewichtiger Einwand, demzufolge auch die raffiniertesten, intelligent wirkenden „Virtuellen Kreaturen" nicht mehr als lediglich elaborierte Simulationen darstellen.

Und doch drängt sich an dieser Stelle und in vielen Literaturquellen immer wieder die Frage auf, ob diese digitale bzw. virtuelle Welt nur ein Segen für die weitere Entwicklung und dem damit verbundenen Megatrend der Individualisierung im privaten und beruflichen Umfeld darstellt oder ob diese auch bisher unerkannte oder nicht berücksichtigte Gefahren und Risiken und damit Nachteile mit sich bringt. Diese Frage soll in den nächsten Unterkapiteln näher untersucht werden.

2.5.1 Chancen und Vorteile durch die Anwendung von Virtual Reality

Neue Umsatzzahlen aus einer Studie von PwC vom Oktober 2019 zeigen, dass die Virtual Reality-Technologie der Nische zu entwachsen scheint um auch zunehmend massentauglich zu werden. So stiegen die Erlöse 2018 in Deutschland bereits auf EUR 116 Mio. (+ 38 % im Vergleich zu 2017). Der größte Anteil entfiel dabei mit EUR 62 Mio. auf den *Gaming*-Bereich (+ 31 % im Vergleich zu 2017). Dahinter folgten mit EUR 43 Mio. (+ 48 % im Vergleich zu 2017) die Erlöse aus dem Verkauf von Virtual Reality-Videos. In der VR-Auskopplung des PwC-„German Entertainment and Media Outlook 2019 bis 2023" ist von einem jährlichen Wachstum von 19 % die Rede.

Virtual Reality bietet zweifellos die Vorteile, dass der Anwender mehr Freiheit besitzt und ortsungebunden arbeiten und interagieren kann. Die zeitliche Komponente tritt stark in den Hintergrund und es bieten sich mehr Gestaltungsspielräume, sowie die Möglichkeit lauten und unpersönlichen Mehrpersonenbüros zu entfliehen.

Zusätzlich bieten sich Vorteile in Bezug auf Qualitätsverbesserungen vor allem in den Bereichen Medizin, der Luftfahrtindustrie, der Fahrzeugtechnik, der Raumfahrt sowie in der Ausbildung und in Schulungen von Anwendern.

In der Medizin findet Virtual Reality beispielsweise zur Qualitätsverbesserung von Koronardiagnostik und -intervention durch Virtual Reality-Simulation Anwendung, denn „die Katheterzahlen in Herzkatheterdiagnostik und -intervention steigen seit Jahren exponentiell an. Hieraus ergibt sich ein enormer Ausbildungsbedarf in der interventionellen Kardiologie. Gleichzeitig haben sich die Zeitressourcen der Krankenhäuser für Fort- und Weiterbildung infolge von Personalmangel und zunehmender Arbeitsverdichtung deutlich reduziert. Eine Lösung dieses Dilemmas sind neue Ausbildungskonzepte, bei denen ein Teil der Ausbildung durch das Training an Modellen und Simulatoren ergänzt oder ersetzt wird. Abhängig von der Komplexität der zu trainierenden Prozedur sind die Anforderungen an den Simulator sehr unterschiedlich und reichen vom einfachen Modell (z.B. für Punktionstraining) bis hin zum Virtual Reality-Simulator mit verlinktem *Full-Scale-Mannequin* (z.B. für ein Teamtraining). Mittlerweile gibt es für Koronarangiographie und -intervention Virtual Reality-Simulatoren, die ein realistisches und praktisches Training in Analogie zu den Flugsimulatoren in der Luftfahrt erlauben.“[9]

2.5.2 Gefahren, Risiken und Nachteile bei Virtual Reality-Anwendungen

Virtual Reality bietet die Nachteile, dass eine größere Abhängigkeit von IT und Internetverbindungen sowie die Notwendigkeit der Überwachung dieser Technologie und Schutz vor Zugriff durch Unbefugte notwendig sind. Aber auf keiner anderen Plattform lassen sich Ängste, lebensecht wirkende Emotionen und Abenteuerlust so stark auf den Anwender übertragen, wie in der Virtual Reality-Technologie und genau damit sind auch Gefahren und Risiken verbunden, die in den nächsten Abschnitten und den nachfolgenden Unterkapiteln teilweise auch anhand von konkreten Anwendungshinweisen analysiert werden.

Im Benutzerhandbuch SM-R322 der Firma SAMSUNG wird beispielsweise bezüglich der Verwendung der SAMSUNG Gear Virtual Reality-Brille empfohlen, dass man eine längerfristige Verwendung vermeiden sollte, da diese vor allem bei Kindern zu einem negativen Einfluss auf die Hand-Augen-Koordination, das Gleichgewicht und die Multitasking-Fähigkeiten führen kann. Kindern unter 13 Jahren ist die Verwendung der VR-Brillen überhaupt untersagt.[10]

Ferner soll es bei 1 von 4.000 erwachsenen Anwendern durch Erlebnisse der virtuellen Realität zu starkem Schwindel, Krämpfen und epileptischen Anfällen oder Bewusstlosigkeit kommen, auch wenn ein Anwender zuvor noch nie einen Anfall hatte, bewusstlos geworden ist oder unter Epilepsie leidet. Diese Krämpfe treten häufig bei Jugendlichen unter 20 Jahren auf. Vor Verwendung der Virtual Reality-Brille wird ebenfalls empfohlen einen Arzt zu konsolidieren, wenn man schwanger oder älter ist, oder unter einer binokularen Sehstörung (z.B. durch einen erheblichen unterschiedlichen dioptrischen Wert beider Augen), Herzerkrankungen oder anderen ernsthaften Erkrankungen leidet.[11]

Aus Sicherheitsgründen sollte die Virtual Reality-Brille auch immer im Sitzen verwendet werden. Man sollte sich dabei auch nicht in der Nähe von anderen Personen, Tieren, Gegenständen, Treppen, Balkonen,

[9] Vgl. Niederlag, Lemke, Lehrach, Petgen, Voelker, Ertl (2014), S. 163 ff.

[10] Vgl. SAMSUNG, Benutzerhandbuch SM-R322 (2015), S. 8.

[11] Vgl. SAMSUNG, Benutzerhandbuch SM-R322 (2015), S. 7.

Fenstern, Möbeln oder anderen Objekten befinden, die man an- oder umstoßen bzw. die hinunterfallen könnten.[12]

Für die angenehme Nutzung der virtuellen Realität ist ein unbeeinträchtigter Bewegungs- und Gleichgewichtssinn des Anwenders erforderlich. Man sollte die Virtual Reality-Brille nicht verwenden, wenn man müde ist, unter Schlafentzug leidet, Alkohol oder Drogen zu sich genommen hat, unter Verdauungsproblemen, emotionalem Stress oder einer Angststörung leidet oder wenn man eine Erkältung, eine Grippe, Kopfschmerzen, Migräne oder Ohrenschmerzen hat, da die Verwendung der VR-Brille die Symptome verstärken kann.[13]

Diese Warnungen zu Sicherheit und Gesundheit sind nur ein Beispiel für die personenbezogenen Gefahren und Risiken, die es bei der Verwendung von VR-Brillen zu beachten gilt und sind auf alle gängigen am Markt erhältlichen Produkte der Hersteller von Virtual Reality-Brillen, in Abhängigkeit von der Qualität der eingesetzten Orientierungs- und Annäherungssensoren, übertragbar.

2.5.3 Motion-Sickness (Kinetose)

Motion-Sickness (*Kinetose*) ist auch als Seekrankheit, Reisekrankheit oder *Gamer-Sickness* und Spielübelkeit bekannt. Sie tritt derzeit sehr häufig und sehr schnell beim Betreten von virtuellen Welten mittels Virtual Reality-Brille auf.

Motion-Sickness wird durch Stresshormone hervorgerufen, die der Körper ausschüttet, wenn das visuelle Erleben nicht mit den restlichen Wahrnehmungen übereinstimmt. Bekannte Symptome sind Schwindel, kalter Schweiß, Unwohlbefinden, Übelkeit, Müdigkeit und Lethargie und Kopfschmerzen. Die Symptome müssen mit dem Abnehmen der Brille nicht zwangsläufig abklingen, sondern können auch nach 24 Stunden in Form von Schwindel oder Unwohlsein noch leicht bemerkbar sein Jeder Mensch reagiert darauf unterschiedlich und einige Anwender kommen bereits nach einer kurzen Eingewöhnungsphase mit der Virtual Reality-Brille gut klar.[14]

Eine zu niedrige Bildfrequenz und eine zu geringe Leistung, die zu einem ruckeligen Ablauf der Bilder führen sowie eine zu hohe Latenz, die zu einem zeitlichen Abstand zwischen einer Kopfbewegung und der visuellen Umsetzung führt, begünstigen das Auftreten von *Motion-Sickness*.

Unter einer Latenz (Verzögerung) versteht man die Verzögerung zwischen der Bewegung des Anwenders und der Reaktion in der virtuellen Realität. Je geringer die Latenz ist, desto realistischer ist die Virtual Reality-Erfahrung.

Beim Eintauchen im sitzenden Zustand in die virtuelle Welt tritt *Motion-Sickness* öfter auf, als bei körperlicher Bewegung in der Virtual Reality mit einer Oculus Rift oder HTC Vive, da dadurch der Konflikt im Gehirn und damit auch die Anfälligkeit für *Visual-Induced-Motion-Sickness* (VIMS) reduziert wird.

[12] Vgl. SAMSUNG, Benutzerhandbuch SM-R322 (2015), S. 9.

[13] Vgl. SAMSUNG, Benutzerhandbuch SM-R322 (2015), S. 10.

[14] Vgl. OnlyVR, Das VR Magazin (2015), Online-Quelle [07.05.2019].

2.5.4 Simulator-Sickness

Die Simulator-*Sickness* ist eine Form der *Motion-Sickness*. Sie beschreibt das Unwohlsein, das durch simulierte Umgebungen (durch künstliche Rotationen und Bewegungen) entstehen kann. Für Simulator-*Sickness* ist hauptsächlich verantwortlich, dass das Auge dem Gehirn sagt, dass man sich in Bewegung befindet, das Gleichgewichtsorgan allerdings meldet, dass man im Moment steht oder sitzt. Es ist also genau andersherum als die Form von *Motion-Sickness*, hat aber die gleiche Auswirkung auf das Gehirn, dass widersprüchliche Signale an das Gehirn gesendet werden.[15]

Andere Internetquellen beschreiben diese Form als *Visual-Induced-Motion-Sickness* (VIMS) dadurch, dass diese durch widersprüchliche Signale der Augen und des Innenohrs hervorgerufen wird. Der im Innenohr befindliche Gleichgewichtssinn meldet demzufolge einen Stillstand, während die Augen eine Bewegung signalisieren. Das Gehirn kann diese widersprüchlichen Meldungen nicht verarbeiten und der Konflikt kann zu einer vorübergehenden Übelkeit beim Anwender führen.

2.5.5 Sucht und Abhängigkeitspotential

Die US-Soziologin Sherry Turkle warnt seit Langem davor, dass die Digitalisierung (Verfügbarkeit von *Big Data*) die Art des Kommunizierens verändert, weil es viel einfacher, bequemer, konfliktärmer und schöner erscheint (siehe dazu auch Abb. 8), als jenes mit realen Menschen im Alltag ist. Die verlockenden Möglichkeiten des Eskapismus bzw. der Realitätsflucht und das damit einhergehende Abhängigkeitspotential und der Sucht-faktor aufgrund der intensiven Immersion, die uns die neue Art des Eintauchens in die virtuelle Welt beschert, sind enorm. Immersion wird als Begriff im Sinne eines intensiven Wahrnehmens bestimmter Situationen verwendet.

Abb. 8: Big Data – Internet of Things (IoT), Quelle: SAS Best Pracitces (2018), Online-Quelle [19.01.2020].

Seit Mitte Mai 2019 berichten nun auch sämtliche Tageszeitungen darüber, dass die Video- oder Online-Spielsucht (*Gaming Disorder*) künftig sogar zum weltweit geführten gültigen Katalog der Gesundheits-störungen gehört. Für die Weltgesundheitsorganisation (WHO) wird ein Spiel- bzw. Onlineverhalten problematisch, wenn ein Anwender mehr als 12 Monate alle anderen Aspekte des Lebens dem Spielen oder dem Aufenthalt in virtuellen Welten unterordnet, wenn er seine Freunde verliert und seine Körperhygiene sogar vernachlässigt. *Gaming Disorder* gehört zu den Impulskontrollstörungen.

Dr. Roland Mader, Leiter der Sektion Sucht am Anton-Proksch-Institut in Wien, beschäftigt sich ebenfalls seit langem mit diesem Thema und meint im April 2019 dazu, dass das Spielverhalten zur Sucht werden kann. Die Anzeichen dafür sind der Verlust der Kontrolle darüber, wie lange ein Anwender spielt bzw. sich in der virtuellen Welt aufhält. Andere Aktivitäten daneben werden vernachlässigt, der Anwender zieht sich

[15] Vgl. Korgel (2018), S. 28.

zurück, Kontakte zu Freunden werden vernachlässigt, Hobbys und Schule leiden und der Betroffene kann auch nicht mehr aufhören sich der Immersion hinzugeben, auch wenn negative Konsequenzen drohen.

Das Wort Immersion, ein aus dem Lateinischen (*Immersio* = Eintauchen) stammender Begriff welches stark durch die Virtual Reality-Technik geprägt wird, stellt aus technologischer Sicht den Prozess und die Art des Interagierens mit der virtuellen Welt dar. Mel Slater und Sylvia Wilbur definierten 1997 in ihrem „A Framework for Immersive Virtual Environments (FIVE): Speculations on the Role of Presence in Virtual Environments“ Immersion als „objektiv festellbaren Detaillierungsgrad der sensorischen Wiedergabetreue einer virtuellen Umgebung“. Immersion ist dementsprechend ausschließlich von der messbaren Qualität der künstlichen Umgebung abhängig.[16]

Präsenz ist die Fortsetzung davon und das Ergebnis einer sehr guten Immersion. Das Ziel von Immersion ist es ein Gefühl der Veränderung der eigenen Wahrnehmung durch die virtuelle Realität zu schaffen. Wenn der Anwender eine virtuelle Welt durch eine Virtual Reality-Brille betritt, dann ist es die Erwartung des Anwenders, dass die veränderte Wahrnehmung deutlich spür- und wahrnehmbar ist. Es müssen ebenso sämtliche menschlichen Sinne dadurch angesprochen werden.[17]

Der Grad der Immersion steigt anhand der Einflüsse, die in der echten Umgebung ausgeschaltet, und in die virtuelle Welt übertragen werden. Je realistischer die Bewegungsübertragung, die Reaktion (Latenz) und die Qualität der Sicht mittels Virtual Reality-Brille übertragen werden, desto höher ist die Immersion, die vom Anwender verspürt wird.

Ob der Sound zum Beispiel in Stereo, in 3D oder in 7.1 (*surround sound*) wahrgenommen werden kann, trägt auch entscheidend zur immersiven Erfahrung bei. Während Immersion in erster Linie den Grad des Eintauchens und des Erlebens betrifft, zielt der Begriff *Präsenz* auf den Zustand des "Dort-Seins" ab. Präsenz ist zwar eng verknüpft mit guter Immersion, jedoch bezieht sich Präsenz auch auf das gesamte Erleben in der virtuellen Welt. Dies erfolgt beispielsweise in Form von Figuren, Gegenständen, Werkzeugen, Maschinen, Anlagen oder dem Betreten von virtuell erzeugten Räumen und wie sich all diese Komponenten ineinander- und zusammenfügen.[18]

Der Phänomenologe und Universitäts-Professor DDr. Thomas Fuchs diagnostizierte bereits im Jahr 2010 aus theoretischer Perspektive eine dysfunktionale Entkörperung der Erfahrung (*Disembodiment*) im Rahmen virtueller Realität. Diese wird als eine Entsinnlichung persönlicher Erfahrung, Phantomisierung der Wirklichkeit und als eine Scheinpräsenz des Menschen charakterisiert, was grundsätzlich der Suchtentwicklung und der nicht-sozialen Einstellung entscheidenden Vorschub leistet. Im quasi-hypnoiden Zustand kann der Anwender die Wirklichkeit von Abbild und Fiktion nicht mehr unterscheiden. Der Anwender ist nicht mehr Zuschauer, sondern als Akteur mit unbegrenzten Bewegungsmöglichkeiten mit dem virtuellen Geschehen verbunden und erlebt in dieser eine magische Wirkung der eigenen Tätigkeiten und das Erlebnis berauschender Omnipotenz. Damit erreicht die Immersion ihren maximalen Grad. Der Computer oder das Smartphone werden als Interaktions- und Kommunikationspartner zu einem

[16] Vgl. Slater, Wilbur (1997), Online-Quelle [11.09.2019].

[17] Vgl. OnlyVR, Das VR Magazin (2015), Online-Quelle [07.05.2019].

[18] Vgl. OnlyVR, Das VR Magazin (2015), Online-Quelle [07.05.2019].

potenziellen Gegenstand der libidinösen Empathie (man denke nur an das zärtliche Streichen über das Display bzw. über den Touchscreen). Die Illusion der eigenleiblichen Bewegung im virtuell erzeugten Raum begünstigt auch die Identifizierung mit Avataren ebenso wie die empathische Interaktion mit virtuellen Personen. Man kann geradezu von einer Einverleibung des virtuellen Raums und Geschehens sprechen.[19]

Gesellschaftskritisch gilt es durchaus auch zu berücksichtigen, dass die Erstellung eines Algorithmus für einen Ingenieur ein interessantes Problem darstellt, das es zu lösen und effizient umzusetzen gilt, aber eine gesellschaftliche, soziale und kulturelle Relevanz dabei gewöhnlich eher in den Hintergrund tritt und wenig mit einfließt.

Es gibt zwei Probleme die sich in diesem Zusammenhang abzeichnen. Erstens, wie werden die Menschen diese – an sich neutrale – Technologie einsetzen und zweitens, welche Folgen wird es haben, wenn sie nicht auf unsere gesellschaftlichen und menschlichen Werte abgestimmt ist?

Im Bereich Gaming und Entertainment stößt Virtual Reality bereits heute auf volle Akzeptanz. Ein weiterer Ansatz ergibt sich im Bereich der Datenwissenschaften, genauer in der Datenanalyse. Hier werden Daten in einem virtuellen, dreidimensionalen Raum visuell aufbereitet, um sie so erlebbar zu machen. Damit Virtual Reality in zunehmend mehr Branchen und in Unternehmensbereichen im industriellen Umfeld breiten Einzug findet, muss sich die geistige Haltung und Einstellung der Agilität dieser Technologie grundlegend anpassen.

Wenn Unternehmen den neuen digitalen Möglichkeiten gegenüber aufgeschlossen und bereit sind, sie anhand einzelner Anwendungsfälle für sich intensiver einzusetzen und zu testen, kann sich die Technologie langfristig weiterentwickeln und ihr volles Potenzial in naher Zukunft auch entfalten.[20]

[19] Vgl. Fuchs (2010), S. 263 ff.

[20] Vgl. PwC Österreich (2019), Online-Quelle [24.11.2019].

3 VIRTUAL REALITY-BRILLEN (VR-BRILLEN)

In den 90er-Jahren kamen die ersten Virtual Reality-Brillen auf den Markt, welche nicht nur für den professionellen Einsatz bestimmt waren. Die Forte VFX1 hat beispielsweise bereits eine Auflösung von 263 x 230 Pixel pro Auge, 256 Farben, ein Sichtfeld von 35,5 ° und ein Gewicht von 1.100 g. Von einem *Highend*-Gerät spricht man zu dieser Zeit bei einer Auflösung von 276 x 372 Pixel pro Auge. Die Virtual Reality-Brillen litten zu dieser Zeit aber noch unter einem starken *Motion-Blur*.[21]

Der Begriff *Motion-Blur* (darunter versteht man eine Bewegungsunschärfe) stammt eigentlich aus der Fotografie und bezeichnet eine auf bestimmte Zonen begrenzte Unschärfe in einem Bild mit bewegten Objekten. Dieser Effekt entsteht durch die Geschwindigkeit des Objektes im Zusammenspiel mit der Belichtungszeit. In Spiele-Anwendungen kommt dieser Effekt bewusst zum Einsatz, besonders häufig natürlich in Spielen mit schnellem *Movement*. Durch *Motion Blur* wird künstlich eine hohe Geschwindigkeit simuliert. Dadurch entstehen beispielsweise in Rennspielen Tunneleffekte. Während die Mitte des Bildschirms oder das fokussierte Objekt scharf gezeichnet werden, verschwimmt die Sicht an den Rändern.[22]

2011 überzeugte Palmer Luckey den Doom-Erfinder John Carmack über die *Crowdfunding* Plattform „Kickstarter“ fast 2,5 Millionen US-$ in die Entwicklung der Virtual Reality-Brille Oculus Rift zu investieren und löste damit einen Boom am Markt für Virtual Reality-Brillen aus. 2014 wurde die Firma Oculus schließlich für ca. rund 2,3 Milliarden US-$ von Facebook gekauft.[23]

Heute bieten auch Hersteller wie Sony, SAMSUNG, Microsoft, Google und HTC gute massenmarkttaugliche und somit kostengünstige Virtual Reality-Brillen (vor allem in der stark wachsenden Nische am *Mainstream*-Markt) an.

Die Virtual Reality (VR)-Brille benötigt man grundsätzlich um in die virtuelle Welt abzutauchen. Dabei handelt es sich um eine Art *Headset*, die den kompletten Augenbereich abdeckt und von außen kein Licht durchlässt. Mit Hilfe der Virtual Reality-Brille werden äußere Reize nahezu komplett ausgeblendet. Die Virtual Reality-Brillen bestehen aus einem hochauflösenden Display mit Vergrößerungslinsen und sensiblen Sensoren, die einen dreidimensionalen Raumeindruck ermöglichen.

Virtual Reality-Brillen werden vor allem bei Videospielern, aber auch bereits immer verstärkter professionell im beruflichen bzw. industriellen Umfeld (z.B. für Produktpräsentationen und Produktkonfiguration, für Weiterbildung sowie für Schulungszwecke, in der Architektur zur Planung und für virtuelle Rundgänge durch ein neues Gebäude, in der Medizin für angehende Chirurgen, die gefahrlos komplexe Eingriffe an virtuellen Patienten üben, oder in der Psychologie wo mit Hilfe der virtuellen Realität individuelle Szenarien geschaffen werden, um Patienten bei der Behandlung von Autismus, Phobien oder posttraumatischen Belastungsstörungen aktiv zu unterstützen) eingesetzt.[24]

[21] Vgl. Korgel (2018), S. 12 – 13.

[22] Vgl. Thöing (2011), Online-Quelle [30.05.2019].

[23] Vgl. Donath (2019), Online-Quelle [25.05.2019].

[24] Vgl. Virtual Reality bei Conrad (2017), Online-Quelle [28.08.2019].

In den folgenden Unterkapiteln wird auf Virtual Reality-Brillen aus Karton bzw. Pappe (die sogenannten *Cardboard*-Modelle), als kostengünstige Alternative zu den Virtual Reality-Brillen mit hochwertigen Kunststoffgehäusen, nicht eingegangen.

3.1 Allgemeiner Aufbau der Virtual Reality-Brille

Beim Kauf einer Virtual Reality-Brille ist genau zu untersuchen welche Anforderungen wie die Art und Dauer der Anwendung und welche Qualität der Komponenten, wie z.B. die Qualität der Sensoren, erforderlich ist. Diese Anforderungen spiegeln sich nicht nur im Preis wieder, sondern können auch maßgebliche Auswirkungen auf die Immersion, die Funktionen der VR-Brille und das körperliche Wohlbefinden des Anwenders mit sich bringen. Unter Immersion oder *Presence* versteht man den Eindruck den der Anwender bekommt, dass er sich physisch tatsächlich am gewünschten virtuellen Ort befindet.

Die Immersion wird perfekt, wenn der Anwender tief in die computergenerierte Welt eintauchen kann. Die Immersion entsteht, wie bereits eingangs angeführt, u.a. durch das stereoskopische Sehen, wenn ein Bild in zwei Teile geteilt und dieses im Gehirn des Anwenders zusammengesetzt wird. So erzeugen minimale perspektivische Unterschiede einen 3D-Effekt (siehe dazu auch Unterkapitel 3.1.3). Die Größe des Entfernungsbereiches innerhalb dessen ein Objekt scharf abgebildet wird, wird als Tiefenschärfe bezeichnet.[25]

Um das, für die Anwendung richtige Produkt auswählen zu können, sollen die folgenden Komponenten einer Virtual Reality-Brille genau geprüft werden.

Eine handelsübliche Virtual Reality-Brille besteht aus den folgenden Komponenten: [26]

- einem stabilen Kunststoffgehäuse
- einem oder zwei Display(s)
- zwei (hochwertigen) Linsen
- einem Gyroskop
- einem Magnetometer
- einem Beschleunigungsmesser bzw. -sensor(en) und
- sonstigen Sensoren (z.B. für das Positionstracking)

3.1.1 Kunststoffgehäuse

Das Kunststoffgehäuse soll grundsätzlich leicht, lichtundurchlässig und stabil sein. Im Gesichtsbereich soll die Virtual Reality-Brille mit einer austauschbaren Polsterung versehen sein. Sie sollte mit beispielsweise verstellbaren Riemen einfach und gut am Kopf zu befestigen sein, um einen guten Sitz für den Anwender zu gewährleisten. Der Tragekomfort und somit die Qualität einer Virtual Reality-Brille wird nämlich durch das Gewicht, die Trageriemen, die Belüftung und die Polsterung bestimmt. Einige Virtual Reality-Brillen

[25] Vgl. Biber (2018), Online-Quelle [17.05.2019].

[26] Vgl. OnlyVR, Das VR Magazin (2015), Online-Quelle [17.05.2019].

verfügen daher bereits über ein hochklappbares Visier. Im Gehäuse befinden sich je nach Anwendungsanforderung ein oder sogar zwei Displays.

3.1.2 Display

Das Display (siehe dazu auch Abb. 9) dient zur Darstellung der virtuellen Oberfläche. Vor dem Display befinden sich zwei Linsen, durch die der Anwender durchschaut. Je nachdem wie gut oder schlecht die Auflösung des Displays und die Qualität der verwendeten Linsen ist, erhält der Anwender ein „gutes oder weniger gutes“ virtuelles Bild.

Abb. 9: Display der Highend Virtual Reality-Brille StarVR, Quelle: MIXED (2018), Online-Quelle [22.05.2019].

Mobile Varianten (z.B. die SAMSUNG Gear VR-Brille) besitzen kein eigenes Display, sondern über spezielle Apps wird über das Smartphone das 3D-Bild erzeugt, indem man das Smartphone vor die Linse des Kunststoffgehäuses schiebt.

3.1.3 Linsen

Beim „natürlichen Sehen“ nennt man das Zusammenspiel von linkem und rechtem Auge stereoskopisches oder doppeläugiges Sehen. Das stereoskopische Sehen basiert darauf, dass beide Augen synchron sehen wobei der Sehwinkel sich je nach Entfernung des betrachteten Gegenstandes ändert. Das menschliche Auge ist somit in der Lage, sowohl weit entfernte Dinge scharf zu sehen, als auch solche, die sich in der Nähe befinden. Der Winkel des rechten und des linken Auges ist normalerweise fast identisch.

Die Fähigkeit der Augen eng nebeneinander liegendes getrennt und scharf zu sehen wird als Auflösungsvermögen bezeichnet und wie bereits eingangs angeführt als Sehwinkel angegeben. Das Gehirn kann daraus die räumliche Beschaffenheit der Objekte berechnen.

Das visuelle Bild bei Virtual Reality-Brillen unterscheidet sich dadurch, dass der Anwender zwei Display-Bilder vor den Augen hat, die sich leicht voneinander unterscheiden. Mit einer speziellen Software wird das

virtuelle Bild für das linke und rechte Auge geteilt, gekrümmt und im Gehirn wieder zusammengesetzt, sodass dadurch die dreidimensionale Sicht möglich gemacht wird, die die virtuelle Welt in weiterer Folge so wirklich erscheinen lassen. Damit man mit einer Virtual Reality-Brille jedoch ein scharfes Bild auch sehen kann, benötigt die Virtual Reality-Brille noch Linsen, denn das Bild ist bei einer VR-Brille nur ca. 5 bis 8 cm vom linken und rechten Auge des Anwenders entfernt.

In dieser Distanz schafft das Auge trotz Akkommodation nicht, das Bild scharf zu erkennen. Unter Akkommodation versteht man die Fähigkeit des Auges, die Sehschärfe aktiv an verschiedene Entfernungen anzupassen. Daher enthält die Virtual Reality-Brille Linsen, die das Display-Bild so brechen, dass es auf der Netzhaut als scharfes Bild abgebildet wird. Um das immersive Eintauchen in die Virtuelle Welt nicht negativ (z.B. durch störende Ränder in der Darstellung) zu beeinträchtigen sind daher hochwertige Linsen empfehlenswert (siehe dazu auch Unterkapitel 2.5.3).

Die beiden Linsen zeigen zwar, wie bereits angeführt, die gleiche Szene, aber aus leicht unterschiedlichem Blickwinkel. Dies hat zur Folge, dass die virtuelle Welt dreidimensional und absolut real erscheint. Wenn sich die Linsen sehr nahe an den Augen befinden, scheint das Blickfeld grenzenlos zu sein. Der Blickwinkel der Augen reicht dann nicht aus, die Ränder der Linsen zu erfassen (weites Sichtfeld ohne erkennbare „Grenzen“).

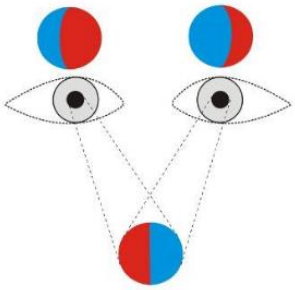


Abb. 10: Stereoskopisches Sehen, Quelle: Conrad (2017), Online-Quelle [30.08.2019].

Abb. 10 zeigt das Prinzip des räumlichen Sehens. Dieser Effekt ist für das Erzeugen einer beeindruckenden Tiefenwahrnehmung in Virtual Reality-Anwendungen notwendig. Je näher sich der betrachtete Gegenstand (in diesem Fall ein rot/blauer Ball) vor dem Auge befindet, desto unterschiedlicher sind die beiden Bilder, welche über die Augen an das Gehirn weitergeleitet werden. Das linke Auge sieht in dieser Darstellung mehr von der roten Fläche und das rechte Auge sieht mehr von der blauen Fläche. Auf Grund der beiden unterschiedlichen Bildinformationen kann das menschliche Gehirn einen räumlichen Gesamteindruck des betrachteten Gegenstandes berechnen.[27]

Die Linsen sorgen auch für die Bildschärfe und leiten im richtigen Einfallwinkel die virtuellen Bilder an die linke und rechte Netzhaut des Auges weiter. Dem Gehirn wird ein Gefühl von Realität und echtem räumlichen Sehen (3D-Effekt) vorgetäuscht. Daher sollen die Linsen, für eine als angenehm empfundene Verwendung, qualitativ hochwertig und einstellbar sein. Ein Sichtfeld zwischen 100 – 120 ° und eine Bildfrequenz von mindestens 60 Bildern pro Sekunde verstärken den Eindruck, dass es sich um eine real wahrgenommene Szene handelt.

Die erste massenmarkttaugliche Generation von Virtual Reality-Brillen startete 2016 (*Release*-Datum) mit einem diagonalen Sichtfeld von etwa 100 °, welches mit dem Sichtfeld durch eine Taucher- oder Skibrille vergleichbar ist. Das natürliche Blickfeld des Menschen, im Vergleich dazu, beträgt ca. 180 °.

Der Anwender hat durch das große Sichtfeld den Eindruck, als würde ihn die virtuelle Welt vollkommen umgeben. Dieser Eindruck entsteht auch dadurch, weil der Anwender sich durch die Virtual Reality-Brille in der virtuellen Welt, indem er den Kopf dreht, auch umschauen kann. Um virtuelle Welten real erlebbar

[27] Vgl. Virtual Reality bei Conrad (2017), Online-Quelle [28.08.2019].

zu machen, müssen sich die Bildinhalte in Abhängigkeit zur Bewegung des Kopfes mitverändern (*Headtracking*).

Optimal für die „ruckel- und schlierenfreie“ Bild-Darstellung sind Bildwiederholfrequenzen von 120 Hz und 240 Hz. An dieser Stelle muss aber auch berücksichtigt werden, dass die Vorteile einer hohen Bildwiederholfrequenz (Anzahl der Einzelbilder pro Sekunde) verloren gehen, wenn das Display zu langsame Reaktionszeiten aufweist.

3.1.4 Gyroskop

Das Gyroskop dient zur exakten Lagebestimmung (durch Angabe der Änderung des auf das Objekt wirkenden Winkelversatzes mit der Zeit), durch das sich Richtungsänderungen genau bestimmen lassen. Bei Virtual Reality-Brillen setzt das Gyroskop die Kopfbewegungen so um, dass es dem Anwender ermöglicht, sich 360° in alle Richtungen zu drehen.

3.1.5 Magnetometer

Unter einem Magnetometer oder *Touchscreens* versteht man einen Sensor, der in der Lage ist ein Magnetfeld zu messen und somit den Bewegungen des Kopfes zu folgen. Er sorgt dafür, dass in der virtuellen Welt die gewünschten Aktionen ausgelöst werden.

3.1.6 Beschleunigungssensoren

Der Beschleunigungssensor wird ebenfalls benötigt, um die linearen Kopfbewegungen zu erkennen und zu übertragen. In einigen Internetquellen werden der Beschleunigungsmesser bzw. -sensor und das Gyroskop unter dem Begriff Orientierungssensor zusammengefasst.

Jedoch sind sowohl Beschleunigungsmesser als auch Gyroskope fehleranfällig, z.B. gegen Rauschen oder gegen Drift (darunter versteht man Positionsfehler, die mit der Zeit auftreten), sodass die Entwickler neue Ansätze zur Erzielung einer optimalen Genauigkeit benötigen. Einer dieser Ansätze ist beispielsweise die Fusion von Sensoren.

3.2 Unterteilung der Virtual Reality-Brillen gemäß ihrer Anwendung

Virtual Reality-Brillen werden grundsätzlich gemäß ihrer allgemeinen Anwendung und den verfügbaren Systemen in vier Kategorien unterteilt. Die Auswahl der richtigen VR-Brille hängt nun davon ab, in welche Kategorie deren Anwendung fällt.

Folgende Parameter sind für die Auswahl der richtigen Kategorie kennzeichnend:

1. Virtual Reality-Brillen, die für den Gebrauch an einem leistungsstarken Computer (z.B. die Oculus Rift VR-Brille, die HTC Vive VR-Brille und viele kleine Hersteller) bestimmt sind.
   Die Interaktion bzw. Navigation erfolgt hier meist mit Handcontrollern. Diese Art der Virtual Reality-Brille verfügt meist über keinen besonders leistungsfähigen Prozessor und die Bilder und Inhalte müssen zusätzlich über ein externes Gerät (externe Sensoren) übertragen werden.

die Virtual Reality-Brille nicht kabellos funktioniert, werden die meisten Geräte über ein HDMI und/oder USB-Kabel an den Computer angeschlossen. Das kann zu Problemen führen, wenn man sich mit der Virtual Reality-Brille viel bewegt (Nachteil: eingeschränkte Bewegungsfreiheit).

2. Virtual Reality-Brillen, die für den Gebrauch mit einer Spielekonsole z.B. für die *PlayStation* 4 (PS4) bestimmt sind. Dazu benötigt man die dazugehörige Virtual Reality-Brille sowie eine PlayStation-Kamera. Wenn die VR-Brille nicht kabellos funktioniert, werden die meisten Geräte auch in dieser Kategorie über ein HDMI und/oder USB-Kabel an die Konsole angeschlossen.
   Das kann auch hier zu Problemen führen, wenn man sich mit der Virtual Reality-Brille viel bewegt (Nachteil: eingeschränkte Bewegungsfreiheit).
3. Virtual Reality-Brillen, die für den Gebrauch mit einem Smartphone (z.B die SAMSUNG Gear VR-Brille) bestimmt sind. Mindestvoraussetzung für die gute Nutzung einer Virtual Reality-Brille mit einem Smartphone sind Android Nougat 7.1 (für die Verwendung des Google Virtual Reality-Headsets „*Daydream*, das bereits über einen Virtual Reality-Modus verfügt“) oder iOS 12.0 (diese bietet eine gute Erkennung von 3D-Objekten) und leistungsstarke Mehrkernprozessoren.
   Die benötigte Größe des Displays wird vom Hersteller der VR-Brille vorgegeben. Je größer das Display ist, desto höher muss auch die Auflösung sein. Für ein 5-Zoll großes Display sollen es mindestens 1.440 x 2.560 Pixel sein, für ein 6-Zoll großes Display bereits 1.440 x 2.880 Pixel. Virtual Reality-Brillen für den Gebrauch mit einem Smartphone bieten den Vorteil, dass man kabellos interagieren kann. Oft liegt die Rechenleistung im Vergleich zu *Gaming*-Computern allerdings weit zurück und diese sind somit weniger leistungsfähig.
4. Schließlich gibt es noch „Autarke Virtual Reality-Brillen“ (diese sind allerdings noch nicht global erhältlich und auch noch recht teuer). Autarke VR-Brillen benötigen keine zusätzliche Hardware. Sie sind mit einer mobilen Recheneinheit (leistungsfähige Prozessoren) ausgestattet und können daher autark und unabhängig verwendet werden.
   Autarke Virtual Reality-Brillen sind die neueste Entwicklung auf dem VR-Brillenmarkt und werden erst seit dem Jahr 2018 bzw. 2019 von Herstellern wie HTC, Lenovo oder Oculus am Markt angeboten.

Aufgrund dieser Kategorisierungsübersicht wird daher für die weitere Ausführung der Anwendungen im Rahmen der Erstellung dieser Masterarbeit die HTC Vive Virtual Reality-Brille ausgewählt, die zudem auch noch ein gutes Preis-Leistungs-Verhältnis aufweist und die technischen Anwendungserfordernisse weitgehendst zu erfüllen vermag.

3.3 Trackingsysteme bei Virtual Reality-Brillen

Bei Virtual Reality-Brillen wird im Wesentlichen zwischen dem *Rotational*- oder *Headtracking* und dem *Positional-Tracking* unterschieden.

Diese werden in den nachfolgenden Unterkapiteln näher analysiert und bewertet.

3.3.1 Headtracking

Das Head- oder *Rotational-Tracking* ist ein *Motion-Tracking*-Verfahren zur Erfassung der exakten Position, Lage, Neigung, Drehung, Rotation, Bewegung und Beschleunigung des Kopfes des Anwenders.

Die Hardwareeinheit bestehend aus Gyroskopen, Magnetometern und Beschleunigungssensoren, die in der Virtual Reality-Brille verbaut sind.

3.3.2 Positional-Tracking

Damit die Position des Kopfes im Raum erkannt und in die virtuelle Umgebung übertragen werden kann, benötigt man zudem noch einen *Positional-Tracker*. Dieser erfasst die genaue Position der Virtual Reality-Brille im Raum und überträgt die Bewegungen naturgetreu in die „Virtuelle Realität".

Dies geschieht entweder über externe Sensoren, oder ein sogenanntes *Inside-Out-Tracking*, bei dem Kameras und Sensoren in der Virtual Reality-Brille die Umgebung und die Bewegungen des Anwenders erkennen. Fehlt ein solches *Positional-Tracking*, kann man in der Virtuellen Welt lediglich seinen Kopf drehen und neigen.

Bewegungen nach oben oder unten, vor und zurück werden dabei nicht erkannt, was zu einer deutlichen Minimierung der Immersion führt und zu *Motion-Sickness* massiv beitragen kann (siehe dazu auch Unterkapitel 2.5.3).

3.4 Bildqualität der Virtual Reality-Brillen

Die Bildqualität der Virtual Reality-Brillen der einzelnen Hersteller wird von Generation zu Generation besser. Neben der Auflösung, der Pixeldichte, der Matrixart und der *Display*größe spielen auch die Qualität der Linsen eine wichtige Rolle. Das Zusammenspiel all dieser zuvor genannten Faktoren beeinflusst, wie stark man noch einzelne Bildpunkte (Pixel) der Virtual Reality-Brille mit dem Auge wahrnimmt. Dieser Effekt wird allgemein auch als *Screendoor-Effect* (Fliegengitter-Effekt) bezeichnet.

Beim Fliegengitter-Effekt entsteht ein feines Gitternetz durch die Abstände zwischen den einzelnen Bildpunkten (Pixeln) des Displays, das den Anwender stark daran erinnert, durch eine Art „Fliegengitter" zu schauen (siehe dazu auch die nachfolgende Abb. 11).[28]

Das liegt vor allem an der Rechnerleistung, denn ein ruckelfreies Bild mit Umrechnung der Kopfbewegungen stellt enorme Anforderungen an den Rechner und an die Grafikkarte.

Techniken wie die *Low Persistence*-Technologie können das Verschwimmen des Bildes bei Kopfbewegungen extrem minimieren. Diese Technologie ermöglicht es, auch in der Bewegung scharfe Bilder in einer Virtual Reality-Brille anzuzeigen. Hierbei wird das Bild nicht dauerhaft auf dem Display angezeigt, sondern es blitzt beispielsweise 60 bis 90-mal pro Sekunde kurz auf, nachdem der *Frame* vom Grafikprozessor berechnet wurde.[29]

[28] Vgl. Korgel (2018), S. 27.

[29] Vgl. Korgel (2018), S. 10.

Die Bezeichnung „2016p“, die beispielsweise bei der PIMAX 8K Virtual Reality-Brille in der nachfolgenden Abb. 11 angeführt ist, bedeutet UHD (*Ultra High Definition*) und steht für eine Auflösung von 3.840 x 2.160 Pixel (Bildpunkte).

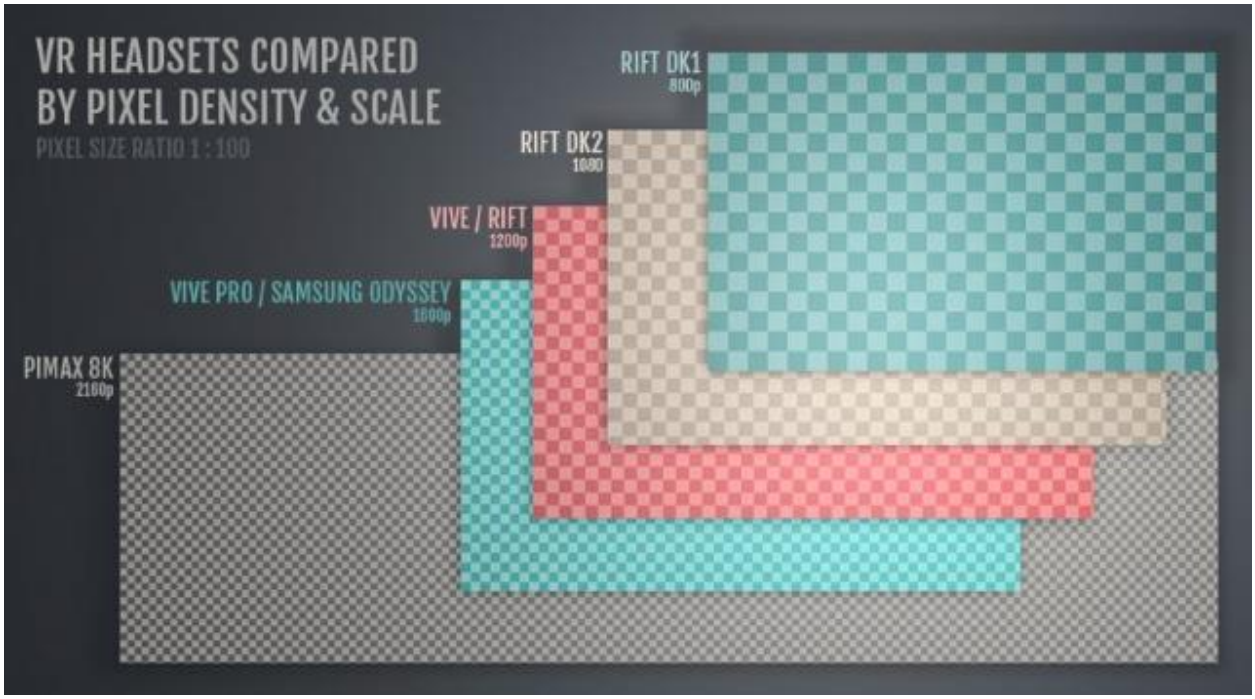


Abb. 11: Bildqualität bei Virtual Reality-Brillen im Überblick, Quelle: VR-NERDS (2018), Online-Quelle [22.05.2019].

Die Anzahl der Bildpunkte (Pixel) ist also ein ungefährer Anhaltspunkt für die Bildqualität der Virtual Reality-Brille. Diese hängt aber auch stark vom Augenabstand zum *Displaypanel* ab. Zudem ist zu beachten, dass durch die Krümmung der Linsen die Pixeldichte in der Mitte des Sichtfeldes der Virtual Reality-Brille höher ist als am Rand.

3.5 Aktuelle massenmarkttaugliche VR-Brillen im Vergleich

2016 kamen mit der HTC Vive, der Oculus Rift, der SAMSUNG Gear sowie der PlayStation VR gleich vier Virtual Reality-Brillen der „ersten Generation“ auf den Markt. 2016 wurde mit der Markteinführung der Virtual Reality-Brillen zusätzlich noch prognostiziert, dass das Interesse und das Potenzial für die Einsatzmöglichkeiten der Virtual Reality-Brillen nahezu endlos sein würden. Dieses Potenzial ist allerdings bis heute noch lange nicht ausgeschöpft.

Vor allem im Gaming-Bereich und im Unterhaltungsbereich (z.B. für Videospiele) stoßen die Virtual Reality-Brillen zwar schon bisher auf ein enormes Interesse und Rufe nach der nächsten Generation werden auch zunehmend stärker aber im industriellen und professionellen Umfeld blieb der Absatz bis heute jedoch weit hinter den kommerziellen Erwartungen zurück.

Globale Wachstumsmarktprognosen geben an, dass der 5G-Standard mit wesentlich schnelleren Übertragungsraten und den gleichzeitig kurzen Reaktionszeiten (*Latenzen*) für einen neuen Aufschwung am globalen Massenmarkt für die Virtual Reality-Brillen-Technologie sorgen könnte.

Immerhin sollen, diesen Prognosen ebenfalls zufolge, im Jahr 2020 weltweit über 50 Milliarden vernetzte Geräte, die Daten produzieren, in Betrieb sein. Das 5G-Netz verbraucht beispielsweise auch nur ein Tausendstel der Energie pro übertragenem Bit im Vergleich zum bisherigen Mobilfunkstandard LTE. Mit 5G werden in der industriellen Anwendung zudem auch z.B. Messdaten mittels Sensorik wesentlich schneller übertragen werden können.[30]

Im Jahr 2019 ist für Virtual Reality-Brillen zudem „die Raumausstattung" (*Room Scale*) von entscheidender Bedeutung. Mit diesem zusätzlichen Feature für diese Virtual Reality-Technologie wird der virtuelle Raum auf den realen Raum geeicht. Erst durch die optimale Abstimmung zwischen diesen entsteht das totale immersive Gefühl, sich auf „natürliche Weise" in der Virtuellen Welt zu bewegen.

Virtual Reality-Anwendungen lassen sich derzeit schon auf einer Fläche von 2 m x 1,5 m ausführen, jedoch sind auch schon Flächen von 6 m x 6 m in realistische Nähe gerückt. Auf diesen Punkt der Raumverfügbarkeit sollte bei der Auswahl der geeigneten Virtual Reality-Brille zusätzlich auch geachtet werden, damit beispielsweise keine Objekte und Gegenstände die Sichtlinie zwischen der Spielfläche und den Sensoren beeinträchtigen.

Die nachfolgenden beiden Tabellen zeigen, die aus Datenblattauszügen aus unterschiedlichen Internetquellen zusammengestellt wurden, einen repräsentativen Überblick ausgewählter Virtual Reality-Brillen der ersten Generation, die in den Jahren 2016 und 2017 im mittleren Preissegment für Endkunden auf den „breiten Markt" gekommen sind.

[30] Vgl. Das Österreichische Industrie Magazin (2019), S. 69.

Bezeichnung	Sichtwinkel diagonal (FOV)	Bildwiederholungs-frequenz (*framerate*)	Beschleunigungsmesser	Gyroskop	Einstellbarer Pupillenabstand Interpupillardistnaz (IPD)	Display	Positionstracking	Auflösung pro Auge
SAMSUNG Gear VR speziell für Smartphones Release 2016	101°	60 Hz Smartphone	ja	ja	Dioptrien An-passung	Smart-phone	nein	1.280 x 1.440 Pixel
Oculus Rift Release 2016	110°	90 Hz	ja	ja	58 – 72 mm	2 x 90 mm OLED	ja	1.200 x 1.080 Pixel
SONY PlayStation Release 2016	100°	120 Hz	ja	ja		5,7 Zoll RGB OLED	ja	960 x 1.080 Pixel
HTC VIVE Release 2016	110°	90 Hz	ja	ja	ja	2 x OLED	Light-house	1.200 x 1.080 Pixel
HOMIDO VR V2 für Smartphone Release 2016	100°	60 Hz Smartphone			Einstellung Kurzsichtig-keit sowie optischer Winkel	Smart-phone: Apple oder SAMSUNG mindestens 4 bis 5,7 Zoll Display	nein	1.280 x 1.440 Pixel

Tabelle 5: Übersicht VR-Brillen, Release 2016, Quelle: Eigene Darstellung.

Bezeichnung	Sichtwinkel diagonal (FoV)	Bildwiederholungs-frequenz (*framerate*)	Beschleunigungsmesser	Gyroskop	Einstellbarer Pupillenabstand Interpupillardistnaz (IPD)	Display (Panel)	Positionstracking	Auflösung pro Auge
LENOVO Explorer VR (Review) Release 2017	ca.110 °	90 Hz	ja	ja		2 x 2,89 Zoll LCD (damit günstiger)	nein	1.440 x 1.400 Pixel
SAMSUNG Odyssey VR Windows Mixed Reality Release 2017	ca.110 °	90 Hz	n.A.	n.A.	60 – 72 mm	2 x 3,5 Zoll AMOLED	n.A.	1.440 x 1.600 Pixel
Acer AH100 VR Microsoft Mixed Reality Release 2017	ca.95 °	90 Hz	n.A.	n.A.		2 x 2,89 Zoll LCD	n.A.	1.440 x 1.440 Pixel
Dell Visor VR Windows Mixed Reality Release 2017	ca.110 °	90 Hz	n.A.	n.A.	Ideal für Brillenträger	2 x 2,89 Zoll LCD	n.A.	1.440 x 1.440 Pixel

Tabelle 6: Übersicht VR-Brillen, Release 2017, Quelle: Eigene Darstellung.

Die durchgängig hohe Bildwiederholungsfrequenz von 90 Hz bei den 2017 erschienenen Virtual Reality-Brillen ist wichtig, da diese hohe Taktrate (*framerate*) für ein flüssiges Bild (ohne Schlieren und Unschärfen bei schnellen Bewegungen) in der Anwendung sorgt.

3.6 Virtual Reality-Brillen im Highend-Bereich

In den folgenden Unterkapiteln werden drei *Highend*-Virtual Reality-Brillen der „zweiten Generation“, die 2018 bzw. 2019 auf den Markt gekommen sind, aus unterschiedlichen Internetquellen analysiert und bewertet. Aus Gründen des Preis-Leistungsverhältnisses bzw. auf Grund der derzeit noch nicht flächendeckenden Verfügbarkeit werden diese VR-Brillen im Rahmen der Erstellung dieser Masterarbeit allerdings nicht weiter betrachtet.

3.6.1 Pimax 5K+ und 8K Virtual Reality-Brille

Die schwedische Firma Starbreeze VR und die chinesische Firma Pimax haben mit ihren Virtual Reality-Brillen die Auflösung auf 5 K bzw. 8 K hochskaliert. Die Pimax 5K+ und die 8K VR-Brillen sind überhaupt erst seit April 2019 praxistauglich.

Das bedeutet, dass es mit dieser Virtual Reality-Brille eine langwierige Herausforderung war, die Bewegungen ohne Latenzen (Verzögerungen) und ohne Verlust der Immersion in die virtuelle Welt zu übertragen (siehe dazu auch Unterkapitel 2.5.5).

Die Abb. 12 und 13 zeigen die nahezu identisch aussehenden Virtual Reality-Brille Pimax 5K+ und 5K-BE, wobei die Zusatzbezeichnung „BE“ für die *Business Edition* steht. Die Pimax 8K verfügt, wie der Name bereits verrät, über eine 8 K-Auflösung. Je nach Wahl bieten diese Virtual Reality-Brillen ein diagonales Sichtfeld (*field of view*) zwischen 125 - 200 °.

Abb. 12: Bildqualität bei Virtual Reality-Brillen im Überblick, Quelle: heise online (2019), Online-Quelle [22.05.2019].

Dieses weite Sichtfeld bietet aber auch den Nachteil, dass es an den Bildschirmrändern zu Verzerrungseffekten kommt, da einige Virtual Reality-Apps für das weite Sichtfeld noch nicht ausgelegt sind. Für gängige Anwendungen empfiehlt sich heute daher eine Sichtfeldeinstellung von 130 bis 150 °. Der sogenannte „Virtual Reality-Tunnelblick“ wird so auch deutlich gemildert.

Alle Pimax-Virtual Reality-Brillen setzen derzeit noch auf das „Valve-Steam-Virtual Reality-Tracking“. Pimax arbeitet aber bereits an einem eigenen Trackingsystem, welches die Hersteller auf den Markt bringen möchten.

Die Pimax-VR-Brille wurde vom chinesischen Hersteller zwar schon für 2017 angekündigt, jedoch zog das Bild 2017 noch kräftige Schlieren und 2018 funktionierte das Positionstracking nicht richtig. Mit dem Release im April 2019 kann man diese VR-Brille nun auch kaufen.

Abb. 13: Deutliche breiteres Sichtfeld der Pimax VR-Brille als die Oculus Rift oder die HTC Vive VR-Brille, Quelle: heise online (2019), Online-Quelle [22.05.2019].

Die auf der „CES“ 2019 vorgestellte Variante der Pimax-Virtual Reality-Brille ist *SteamVR*-kompatibel, zeigt eine gute Bildqualität, einen geringen Fliegengitter-Effekt und es sind keine Verzögerungen mehr zu spüren. Sie weist aber ein leichtes „Wobbeln“ bei schnellen Kopfbewegungen im peripheren Sichtfeld auf. Die *God-Rays*, also die „Weißeinstrahlungen“ bei hellen Bildern auf dunklem Grund (siehe auch Abb. 14), wie sie die Oculus Rift und HTC Vive (Pro) zeigen, können bei der Pimax-Virtual Reality-Brille nun auch wesentlich verbessert werden. Oft bemängelt werden noch die mangelhafte Verarbeitungsqualität, der Tragekomfort und die verbesserungswürdige Software-Qualität.[31]

Abb. 14: Strahlenbüschel (God-Rays) im Kloster Maulbronn, Quelle: fotopraxis (2015), Online-Quelle [25.11.2019].

Strahlenbüschel bzw. auch als *God-Rays* bezeichnet, sind ein Phänomen der atmosphärischen Optik und sind in der Natur ein eindrucksvolles Phänomen, dass dadurch entsteht, dass die direkte Sonnenstrahlung durch die Wolken blockiert wird (Helligkeitskontraste) und Wasser- und Staubpartikel in der Luft diese Strahlen mit einem kleinen Winkel streuen und diese dadurch in das Auge des Beobachters treten. Der gleiche Effekt zeigt sich z.B. auch bei Autoscheinwerfern, wenn der Fahrer durch Nebel fährt.

Die 5K-BE-Version (darunter versteht man, wie bereits eingangs erwähnt, die *Business Edition*) verfügt standardmäßig über ein OLED-Display und die 5K+-Variante über ein LC-Display. Der Unterschied zwischen den beiden Displays liegt darin, dass die OLED-Variante kräftigere Farben und ein satteres Schwarz zeigt als die LCD-Variante.

Das weite Pimax-Sichtfeld und die hohe Auflösung von 2.560 x 1.440 Bildpunkten (Pixel) pro Auge bei der 5K+ sowie 3.840 x 2.160 Bildpunkten (*Pixel*) pro Auge bei der 8K erfordern aber auch eine wesentlich

[31] Vgl. ComputerBase (2019), Online-Quelle [25.11.2019].

höhere Prozessorgeschwindigkeit und damit beispielsweise eine Grafikkarte wie die nVidia GeForce GTX 1080 TI Gaming X. Der Preis dieser Grafikkarte liegt im hohen Preissegment (in Abhängigkeit vom Hersteller liegt diese 2019 im Bereich von ein- bis zweitausend Euro).

3.6.2 StarVR One (XT) Virtual Reality-Brille

Während der chinesische Hersteller Pimax eine neue Generation der Virtual Reality-Brille 2019 auf den Markt bringen konnte, hat das schwedische Unternehmen Starbreeze VR seine Geschäftsideen nach einer drohenden Insolvenz im Dezember 2018 fallengelassen und konzentriert sich zukünftig wieder auf sein Kerngeschäft, nämlich der Spieleentwicklung (*Games*).

Für Geschäftsbereiche, die nicht zum Kerngeschäft *Games* gehören, will das Unternehmen ab sofort mit externen Partnern kooperieren. Das dürfte wohl bedeuten, dass die *Highend*-Virtual Reality-Brille StarVR One in der neuen Unternehmensstrategie keine Rolle mehr spielen wird. Der Verkauf dieser Profibrille startete zwar im November 2018, wurde aber mittlerweile vom Hersteller gestoppt. Die StarVR war ein 2016 von Starbreeze und dem Hardwarehersteller Acer gegründetes Unternehmen, an deren weiterer Fortsetzung auch Acer wenig Interesse zeigte.[32]

Die StarVR One VR-Brille überzeugt laut MIXED vor allem durch ihre Linsen, die große Sichtfeldweite und das klare und verzerrungsfreie Bild. Allein an den Linsen hatte Acer 2,5 Jahre gearbeitet.

StarVR One unterstützt *Steam*VR-Anwendungen und verfügt über ein großes Sichtfeld von 210 ° diagonal und 130 ° horizontal. Der Vorteil dieses weiten Sichtfeldes ist es, dass sich der Anwender weniger mit dem Kopf dreht und mehr die Augen (*Eye-Tracking*) einsetzt. Die StarVR One Virtual Reality-Brille verfügt über ein hoch-auflösendes AMOLED-Display mit einer Auflösung von 1.830 x 1.464 Pixel pro Auge, eine Bildwiederholfrequenz von 90 Hz sowie über eine volle RGB-Matrix. Diese Virtual Reality-Brille liegt in einer sehr hohen Preisklasse von mehreren Tausend Euro und damit wird klar, dass diese VR-Brille sich nicht an Endkunden richtet.

Die automatische IPD-Einstellung (damit versteht man den einstellbaren Pupillenabstand) mittels *Eye-Tracking* zeichnen diese Virtual Reality-Brille aus. Zwei Youtuber konnten auf der IFA 2018 diese VR-Brille testen und stellten fest, dass im Sichtfeld keine Ränder mehr zu erkennen waren und dass keine störenden Lichtreflexionen bei dieser Virtual Reality-Brille mehr auftreten, wie sie bei der Oculus Rift und der HTC Vive beispielsweise noch zu sehen sind.

Der VR- und AR-Analyst Anshel Sag schrieb für die Webseite „Forbes“ auch, dass diese Virtual Reality-Brille mit ihren 16 Millionen RGB-Subpixel keine Fliegengitter-Effekte mehr zeigt. Als störend wird lediglich die ungleiche Ausleuchtung (die sogenannte „*Mura*“) genannt.

Neben der Virtual Reality-Brille versucht Starbreeze sich am Aufbau eines VR-Arcade-Netzwerks, einer VR-Spielhalle in Los Angeles um erstklassige VR-Erlebnisse für die breite Masse zugänglich zu machen, und bietet dafür das volumetrische Filmformat “PresenZ” an.

[32] Vgl. MIXED, Das Online-Magazin für Mixed Reality (2018), Online-Quelle [22.05.2019].

3.6.3 5K Virtual Reality-Brille XTAL

Die XTAL Virtual Reality-Profibrille wurde im Juni 2018 vom Prager Startup Unternehmen Vrgineers für Unternehmen und professionelle Anwender im industriellen Umfeld folgendermaßen vorgestellt:[33]

„Die XTAL VR-Brille verfügt über zwei OLED-Displays, die mit 2.560 x 1.440 Bildpunkten (Pixel) auflösen. Die beiden asphärischen Linsen wurden eigens für diese VR-Brille entworfen und haben im Gegensatz zu den meisten anderen VR-Brillen keinen Fresnel-Schliff. Dadurch sollen keine störenden Lichtreflexionen auftreten, wie man sie von HTC Vive und Oculus Rift immer noch kennt. Das diagonale Sichtfeld liegt bei 170 Grad.

Diese Virtual Reality-Brille unterstützt *Eye-Tracking* und passt den Abstand der Linsen automatisch an den Augenabstand des Anwenders an. Eine weitere Besonderheit dieser Virtual Reality-Brille ist das vollintegrierte *Leap-Motion*-Modul, das Hände und Finger des Nutzers erfasst und in die Virtual Reality überträgt. Das Gerät erkennt zudem auch noch Sprachbefehle."

Abb. 15: Das Leap-Motion Modul der VR-Brille XTAL, Quelle: MIXED (2018), Online-Quelle [22.05.2019].

In der Abb. 15 sieht man wie der Anwender mittels Handbewegungen und Sprachbefehlen mit dem 3D-Modell eines Fahrzeugs interagiert: Mit den entsprechenden Handbewegungen öffnet er die Autotüren, wechselt die Gänge und bedient das Lenkrad. Mit seinen Sprachbefehlen ändert er die Farbe der Lackierung und die Art der Felgen. Die Anzeigen auf dem Tachometer sind gut ablesbar und selbst feine Nahtstellen in der Polsterung sind für den Anwender erkennbar. Dieses *Highend*-Gerät bringt klare Vorteile hinsichtlich der Produktvisualisierung im industriellen Umfeld mit sich.

2018/19 will das Startup-Unternehmen außerdem ein Update veröffentlichen, das die Möglichkeit bietet *Inside-Out-Tracking* zu aktivieren und zusätzlich ein Mixed Reality-Kameramodul herausbringen, das die Außenwelt aufnimmt und dann auf die Displays der Anwendungen *streamt*.

Diese Virtual Reality-Brille lag bei der Markteinführung in einem sehr hohen Preissegment von immerhin mehreren Tausend US-$.

[33] Vgl. Bezmalinovic (2018), Online-Quelle [22.05.2019].

3.7 Virtual Reality-Ganzkörperanzug (HoloSuit)

Der Virtual Reality-Ganzkörperanzug (auch bezeichnet als *HoloSuit*) ist eine interessante Ergänzung zu den gängigen Virtual Reality-Brillen (Virtual Reality-*Headsets* oder *Head-Mounted Displays*), die es derzeit bereits in der „zweiten Generation“ für die Anwender am Markt zu kaufen gibt.

Dieser Anzug zeichnet sich durch volle Bewegungsfreiheit für den Anwender aus, ohne über 3D-*Capture-Devices* wie externe Sensoren zu verfügen. Er bietet auch den Vorteil, dass kein externes Trackingsystem für diesen Anzug mehr benötigt wird. Damit muss man auch nicht auf die klassischen *Input-Devices,* wie beispielsweise ein *Keyboard*, die Computermaus, Virtual Reality-Controller, etc., für die Interaktion in der Virtuellen Welt mehr zurückgreifen (siehe dazu auch die nachfolgenden Abb. 16 und 17).

Das Südindische Unternehmen Kaaya Technology Inc. finanzierte die Entwicklung dieses waschbaren HoloSuits über die Crowdfunding-Plattform „Kickstarter“ und brachte diesen Anzug 2018 für Endkunden auf den Markt.

Ziel dieses Produktes ist es, dass durch Ganzkörperbewegungen die Arbeitsabläufe (beispielsweise für das Erlernen von spezifischen Fähigkeiten und Fertigkeiten im Ausbildungs- und Trainingsbereich) besser in das Muskelgedächtnis des Anwenders übergehen sollen.[34]

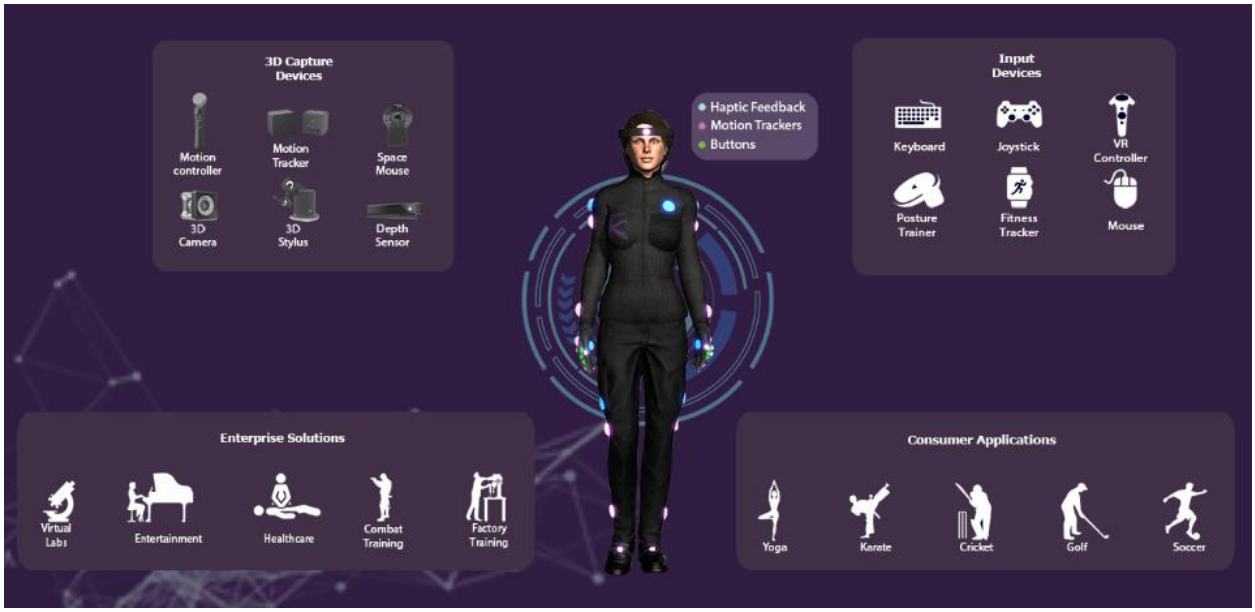


Abb. 16: Der HoloSuit Ganzkörperanzug, Quelle: HoloSuit (2018), Online-Quelle [27.05.2019].

[34] Vgl. VRPlayground (2018), Online-Quelle [19.01.2020].

Der HoloSuit ist derzeit in den folgenden zwei Ausführungen erhältlich:[35]

HoloSuit Pro mit

- 36 Sensoren für die komplette Bewegungserkennung (verteilt auf 2 Handschuhe, 1 Hose, 1 Schuhaufsatz, 1 Jacke samt Stirnband wie in den Abb. 16 und 17 dargestellt).
  Die Sensoren werden mittels Klettband am Anzug angebracht und können abgenommen werden.
- 9 Motoren für haptisches Feedback (an den Schultern, an der Brust, an den Händen, an den Oberschenkeln und am Steißbein)
- 6 Bedienelementen.

Das komplette Set kostete 2018 bei der Markteinführung rund zweitausend Euro.

HoloSuit Standard mit

- 26 Bewegungssensoren (verteilt auf 1 Jacke oder Weste samt Kopfbedeckung, 2 Handschuhe und 1 Hose mit Fußbedeckung)
- 9 Motoren für haptisches Feedback (an den Schultern, an der Brust, an den Händen, an den Oberschenkeln und am Steißbein)
- 6 Bedienelementen.

Das komplette Set kostete 2018 bei der Markeinführung rund eintausend Euro.

Abb. 17: Der HoloSuit Ganzkörperanzug, Quelle: HoloSuit (2018), Online-Quelle [27.05.2019].

Der Vorteil dieses Virtual Reality-Anzuges ist die Ganzkörpererfassung von Bewegungsdaten in Virtual Reality-Anwendungen, die sich positiv auf das Präsenzgefühl des Anwenders auswirkt.

In der nachfolgenden Abb. 18 sieht man die Verteilung der Bewegungssensoren auf dem HoloSuit Virtual-Reality Ganzkörperanzug. Auf den Handschuhen verdichten sich die Anzahl der Sensoren von 20 der insgesamt 36 Sensoren.

Die Bewegungsdaten werden über WLAN bzw. Bluetooth LE über dem Computer an die Spieleengines Unity 3D oder Unreal bzw. an Android, iOS oder Windows gesendet. Hier können schließlich über die „*Unity SDK*“ (siehe dazu auch Kapitel 11) die Bewegungsdaten auf den virtuell erzeugten Avatar übertragen werden.

[35] Vgl. VRPlayground (2018), Online-Quelle [27.05.2019].

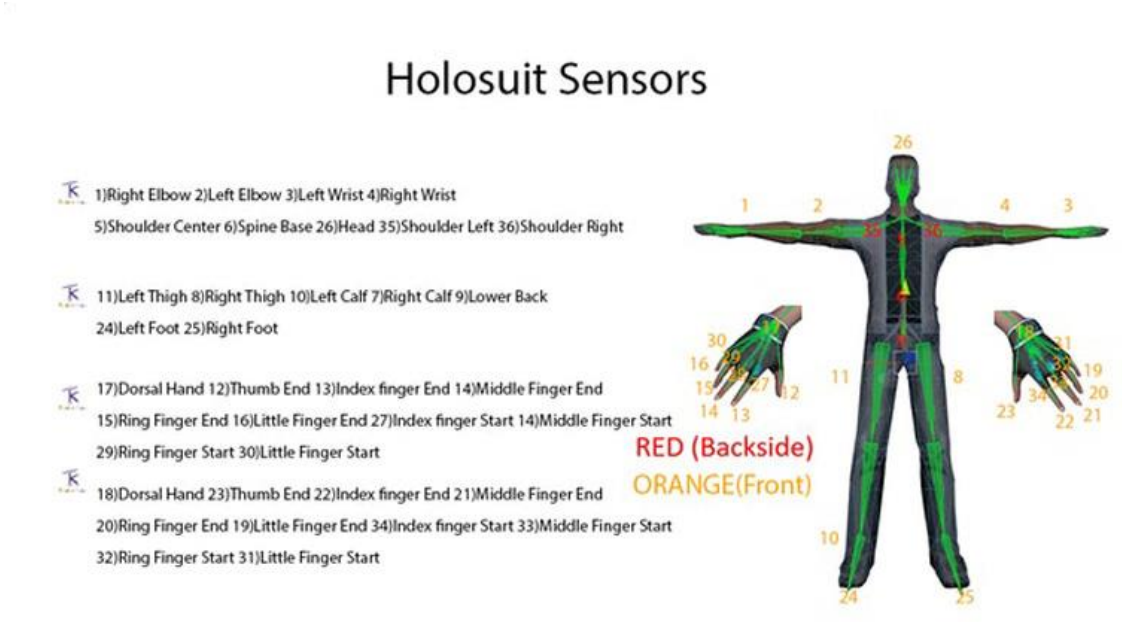


Abb. 18: HoloSuit Bewegungssensoren, Quelle: VRPLAYGROUND (2018), Online-Quelle [27.05.2019].

In der nachfolgenden Abb. 19 ist die Verteilung der haptischen Feedbacks dargestellt. Haptisch bedeutet, dass als Bestätigung (*Feedback*) der erfolgreichen Dateneingabe bzw. –erfassung, ein Impuls (z.B. Vibrationen, Stromimpulse) an den menschlichen Körper abgegeben wird und der Anwender diesen dann z.B. am Handrücken spürt.

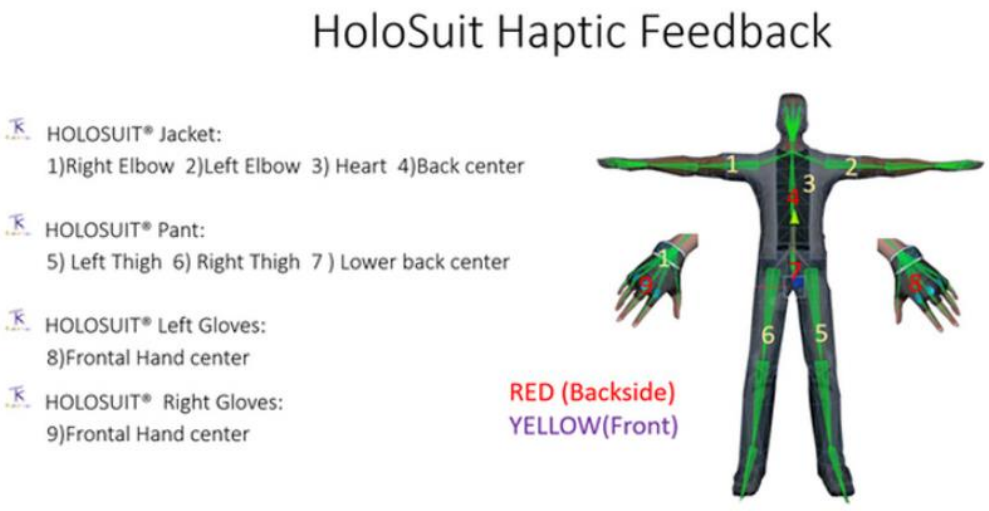


Abb. 19: HoloSuit – haptisches Feedback, Quelle: VRPLAYGROUND (2018), Online-Quelle [28.05.2019].

Die breiten Anwendungsgebiete für den Virtual Reality-Ganzkörperanzug sind vor allem die Steuerung eines U-Bootes beim Militär, im Zuge eines Feuerwehreinsatzes, die Fernsteuerung eines Roboters, im Sportbereich, für Spiele oder *Motion-Capturing*.

Der HoloSuit ist mit Virtual Reality-Brillen wie der HTC Vive, Oculus Rift, SAMSUNG Gear, der Microsoft HoloLens Mixed Reality und Windows Mixed Reality Brille aber auch mit iOS und Android kompatibel.

Im Oktober 2019 konnten Wissenschaftler des Helmholtz-Zentrums in Dresden-Rossendorf (HZDR) rund um Dr. Densys Makarov und der Johannes-Kepler-Universität in Linz nun erstmals auch einen elektronischen Sensor entwickeln, der gleichzeitig Reize sowohl berührungslos als auch durch direkten Kontakt verarbeiten kann. Bisherige Systeme funktionierten entweder nur, indem sie tatsächlich physische Berührungen registrieren oder indem sie Objekte über technische Mittel berührungslos verfolgen.

Zum ersten Mal nun sind die beiden Interaktionswege nun auf dem Sensor, den die Wissenschaftler als „Magnetisches mikroelektromechanisches System“ (m-MEMS) bezeichnen vereint. Dafür haben Wissenschaftler auf einer hauchdünnen Polymer-Folie einen Magnetsensor angebracht, der nach dem Prinzip der Änderung des elektrischen Widerstandes arbeitet. Diese Folie verschließt ein Loch, das genau in der Mitte einer zweiten Silizium-basierten Polymerschicht (Polydimethylsiloxan) liegt. In diese Aussparung wird ein Permanentmagnet eingefügt, aus dessen Oberfläche weiche, pyramidenartige Spitzen herausragen. Da sich der Sensor problemlos auf der menschlichen Haut auftragen lässt, ist es möglich, intuitivere und natürlichere Interkationen in Umgebungen der Virtuellen oder Erweiterten Realität durchzuführen.[36]

[36] Vgl. pro-physik.de (2019), Online-Quelle [20.01.2020].

4 3D-GAME-SOFTWARE (3D ENGINES)

Das Entwickeln von Programmen über Virtual Reality Engines ist ein relativ junges Thema. Bei 3D-Engines spricht man im Wesentlichen von der Unity 3D 2018 und Unreal, die eine einfache Entwicklung von innovativen Virtual Reality-Anwendungen sowie von Spielen ermöglichen. Unity 3D hatte 2018 einen Marktanteil von ca. 48 % und die Unreal Engine lag im Vergleich dazu bei ca. 13 %. Die restlichen Marktanteile von 39 % verteilen sich auf die Frostbite Engine von Electronic Arts, die Cry Engine fon Crytek, die Anvil Engine von Ubisoft und die Source Engine von Valve.

Die Auswahlentscheidung des Einsatzes der richtigen Software ist von den Projektanforderungen und von zahlreichen anderen Faktoren abhängig, die in den folgenden Unterkapiteln näher analysiert werden sollen, um damit die Auswahl der geeignete 3D-Game-Engine zu ermöglichen.

4.1 Spieleengine Unity 3D 2018

Unity ist eine Multiplattform 3D- und 2D-Spieleengine, die von Unity Technologies in Dänemark entwickelt wurde. Als Multiplattform kann man für Windows Computer, Mac, Linux, Smartphones, Browser und Videospielkonsolen Anwendungen entwickeln. Unity verwendet die Programmiersprache C# und es ist auch notwendig, einige Code-Zeilen für die Entwicklung einer Anwendung selber zu schreiben. Die Unity Engine unterstützt eine große Anzahl an verschiedenen Virtual Reality-Brillen, ohne dass man die Brille selbst in der Entwicklung und in der virtuellen Anwendung integrieren muss.

Abb. 20: Unity Logo, Quelle: Unity Technologies (2019), Online-Quelle [25.05.2019].

Unity wird mit drei unterschiedlichen Lizenzen „Personal“ (kostenlose Version für die derzeit nichtkommerzielle Anwendungen für den Einstieg), „Plus“ (diese Lizenz wird von Unternehmen mit einem Umsatz größer US-$ 100.000,-- / Jahr benötigt; die Kosten betragen US-$ 35,-- / Monat) und „Pro“ (für Onlinespiele, die Kosten betragen US-$ 125,-- / Monat) angeboten. Für die Nutzung der Lizenz „Personal“ ist die Bestätigung „*I dont’t use a professional capacity*“ im Zuge der Installation erforderlich. Projekte kann man auch in die *Cloud* (*In The Cloud*) hochladen um mit anderen Anwendern interaktiv zu arbeiten oder um den *Sourcecode* teilen zu können. Der Anwender kann auch direkt auf der Festplatte *(On Disk*) arbeiten.

Der Einstieg in diese Engine ist auch für Nichtprogrammierer relativ leicht möglich. Der Anwender kann ein *Plugin* für Microsoft Visual Studio nutzen, mit welchem sich Unity dann verknüpft und man programmiert dann direkt in *Virtual Reality-Microsoft Visual Studio*. Dies erfolgt über die Auswahl der Komponente „Microsoft Visual Studio Community 2017“ direkt beim Installieren von Unity 3D 2018. Es ist auch erforderlich eine Unity-ID (Benutzerkonto) zu erstellen um auf die Unity *Services* zugreifen zu können. Dazu zählt auch der Zugriff auf den *Asset Store* und auf das offizielle Unity-Forum. Beim ersten Starten von Unity ist diese Unity-ID dann einzugeben und für die anschließenden *Einlogging*-Vorgänge zu speichern (*remember me function*).

Ein neues Projekt (*new* in der Projekt-Auswahl) kann man auch mit einer Hilfsfunktion von Unity (*Enable Unity Analytics*) erstellen. Wenn man standardmäßige *Assets* aus Unity einfügen möchte, kann man auf *Add Asset Packages* klicken um diese dann auch schon direkt z.B. zum Testen zu nutzen (z.B. *Characters*).

Der Fokus bei Unity 3D 2018 liegt auf *Plug & Play*. Im Unterschied zur Unreal Engine ist die Grafik bei Unity etwas einfacher und die Detailmöglichkeiten sind etwas eingeschränkter. Vorsichtig sollte man beim Update von Unity auf neuere Versionen sein. Es ist vor der Durchführung des Updates unbedingt erforderlich sicherzustellen, dass ältere Versionen mit der neueren Version noch kompatibel sind und alle notwendigen Features auch noch unterstützt werden.

Eine weitere Möglichkeit zur Programmierung von virtuellen Anwendungen ist „*WebVR mit A-Frame*“. Hier benötigt man für das Entwickeln einer virtuellen Anwendung allerdings schon recht gute JavaScript- und HTML-Kenntnisse.

4.2 Unreal Engine

Abb. 21: Unreal Logo, Quelle: unrealengine.com (2019), Online-Quelle [18.11.2019].

Die Unreal Engine 4 wurde im Mai 2012 von *Epic Games* veröffentlicht und ist bereits als Entwicklungsumgebung konzipiert. Sie ist der direkte Nachfolger des *Unreal Development Kit* (UDK), die von Programmierern nur eigenschränkt eingesetzt werden konnte. Für beispielsweise einfache oder experimentelle Zwecke kann die Unreal Engine der Unity 3D Engine sehr einfach vorgezogen werden. Die Vollversion der Unreal Engine ist kostenlos erhältlich, Voraussetzung dafür ist allerdings, dass der Umsatz des Anwenders weniger als 3.000 US-$ / Monat beträgt.

Die Benutzeroberfläche der Unreal Engine ist der Unity 3D Engine sehr ähnlich, jedoch ziemlich aufgebläht und komplex und man benötigt definitiv für die Entwicklung einer Anwendung wesentlich mehr Zeit als bei Unity 3D. Das Importieren und Speichern von *Assets* dauert länger als bei Unity 3D und selbst einfache Aufgaben erfordern zahlreiche zusätzliche Schritte. Die Benutzeroberfläche von Unity 3D ist im Vergleich dazu sehr reaktionsschnell. Unter *Assets* versteht man alle möglichen Ressourcen, die der Anwender in einem Projekt verwendet (*Textures*, *Materials*, *Sounds*, *Scenes*, Prefabs, 3D-Modelle, etc.)

Im April 2019 erschien mit einem Mega-Update die Version 4.22 der Unreal Engine. Mit dem Update hat der Hersteller dieser Spiele-Engine den Schwerpunkt merklich auf Geschwindigkeit, Kollaboration und Responsivität gelegt. Integriert sind mit Unreal Engine 4.22 ab jetzt unter anderem *Real-Time-Raytracing*-Funktionen. Mit dieser Funktion lassen sich aus einer Reihe von *Shadern* und Effekten realistisch wirkende Lichteffekte in Echtzeit erzielen (siehe dazu auch Abb. 23), sowie

Abb. 22: Darstellung eines Bauteiles in der Unreal Engine 4.22, Quelle: UnrealEngine (2019), Online-Quelle [18.11.2019].

ein neues *Live-Coding*, damit lassen sich in C++ Änderungen in der Entwicklungsumgebung vornehmen, innerhalb von wenigen Sekunden kompilieren und in einen laufenden Editor oder eine eigenständige Anwendung umwandeln.[37] Des Weiteren sind Echtzeit *Compositing* (mit präziser Kantenverarbeitung, eine optimierte Farbtrennung, natürliche Farbtreue und verbesserte Farbsaumunterdrückung), ein *Multi User Editing* (*Live*) (mit diesem soll die Zusammenarbeit mehrerer Mitglieder verbessert und vor allem direkter gestaltet werden können), und OCIO (*OpenColorIO*) Farbprofile in der Unreal Engine integriert.

Die Anpassung *OpenColorIO* wurde zur Unterstützung von Workflows entwickelt, die auf eine OpenColorIO-Farbverwaltung setzen. Mit dieser Anpassung kann der Anwender die Farben zwischen Quell- und Zielfarbräumen umwandeln.

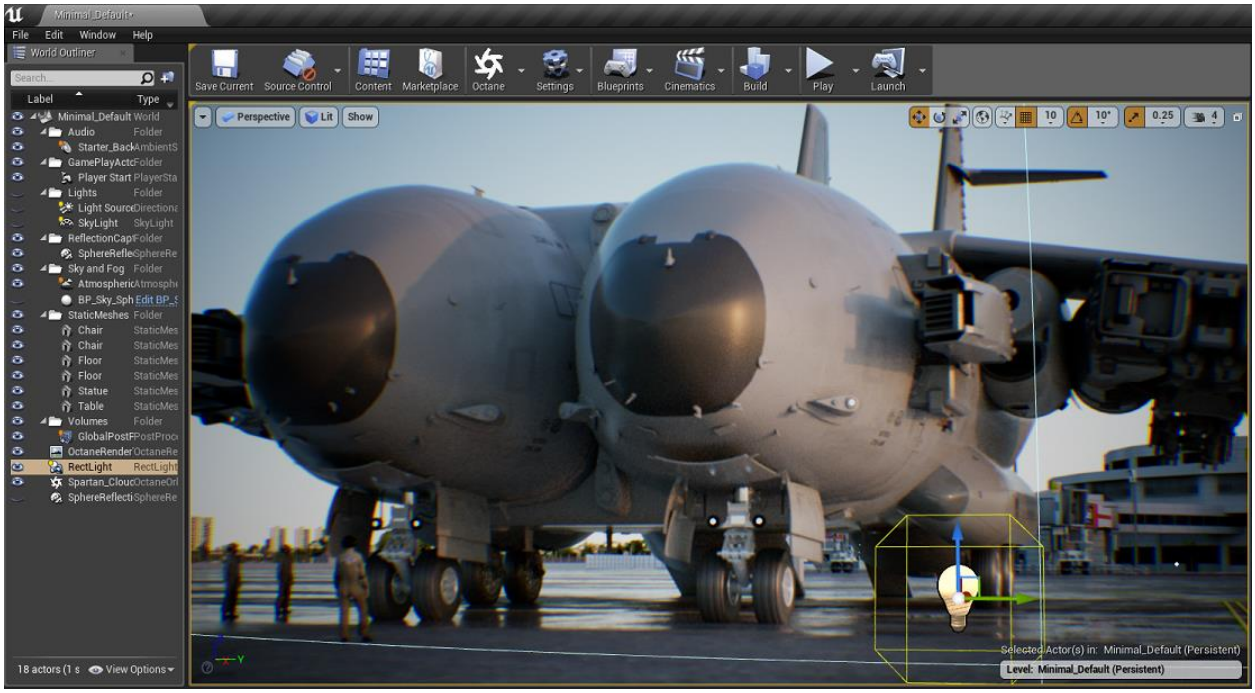


Abb. 23 Darstellung von Reflektionen zur Simulation von akkuraten Umgebungs-Reflektionen auf reflektierenden Flächen in der Unreal Engine, Quelle: Digital Production (2019), Online-Quelle [18.11.2019].

Für das Arbeiten mit der Unreal Engine wird vom Hersteller Windows 10 RS5-Update, DirectX 12 und DirectX Raytracing DXR sowie eine leistungsstarke NVIDIA GeForce RTX Grafikkarte mit den neuesten Treibern empfohlen. Wenn es um die Grafik geht ist die Unreal Engine der Unity 3D weit überlegen. Von komplexen Partikelsimulationssystemen bis hin zu fortschrittlicher Beleuchtung bietet die Unreal Engine viele Features für den Anwender an.[38]

Da die Unreal Engine jedoch auch wesentlich komplexer und aufwendiger in der Erstellung und Anwendung von Programmcodes ist als Unity 3D 2018, soll in der Umsetzung des praktischen Teiles dieser Arbeit die Unity 3D Engine eingesetzt werden. Bei der Unreal Engine steht für Einsteiger ohne tiefe Programmierkenntnisse, für das Schaffen einer Anwendung, außerdem „*Blueprint*“ zur Verfügung.

[37] Vgl. Tom Jansen, Digital Production (2019), Online-Quelle [11.11.2019].

[38] Vgl. Viscircle (2018), Online-Quelle [11.09.2019].

5 VISUALISIERUNG UND VIRTUELLE STEUERUNG EINES DIGITAL TWIN-MODELLES IN HARD- UND SOFTWARE

Für die Ein- und Ausgabe der erforderlichen Informationen (Messdaten) soll ein einfaches Ein- und Ausgabe-Board mit einem Mikrocontroller und analogen Ein- und Ausgängen verwendet werden. Die Programmierung soll in einer leicht zu erlernenden Programmiersprache in C bzw. C++ erfolgen und umfangreiche Bibliotheken und Beispiele sollen die Programmierung vereinfachen.

Aus diesem Grund wird als Eingabe/-Ausgab-Board ein Arduino kompatibles Nano-Board, Version V3.0 für die Ein- und Ausgabe der Messdaten ausgewählt. Arduino bringt eine eigene integrierte Entwicklungsumgebung (IDE) mit. Es hat außerdem den großen Vorteil gegenüber herkömmlichen geschlossenen Messgeräten, klein, kostengünstig, kompakt und dennoch transparent und leistungsfähig zu sein.

Für die Programmerstellung und -ausführung in der Entwicklungsumgebung „Arduino IDE“ (Version 1.8.4) und in der Spieleentwicklungssoftware Unity 3D 2018 (Version 2018.1.6f1), sowie für die Durchführung der Tests im Zuge der Umsetzung des praktischen Teiles dieser Masterarbeit, soll demnach ein Arduino kompatibles Testboard mit einem USB-Chip eines chinesischen Anbieters eingesetzt werden. Dafür muss auch berücksichtigt werden, dass ein geeigneter Treiber (CH340) gefunden und installiert werden muss, damit das kompatible Board unter Windows auch zuverlässig erkannt und die Schnittstellenfunktion gewährleistet wird.

Die wesentlichen Ziele für die Visualisierung und Steuerung des Digital Twin-Modelles sind das Messen der aktuellen Umgebungstemperatur in der realen Welt, das Messen der aktuellen Luftfeuchtigkeit in der realen Welt, das Anzeigen dieser Messwerte in der virtuellen Welt und das gezielte Ansteuern von vier realen Leuchtdioden über die virtuelle Welt sowie schließlich die Programmierung einer weiteren Schnittstelle für die Ausgabe von Messdaten über Modbus TCP, weil damit die wesentlichen Funktionen dieser Schaltung visualisiert und für die weitere Analyse bereitgestellt werden können.

Über eine serielle Schnittstelle (COM-Port) soll der Datenaustausch zwischen dem Arduino kompatiblem Board und dem Computer erfolgen. Das *BIOS* und das Betriebssystem sollen dabei einer physisch existierenden Schnittstelle mit COM1, COM2, usw. eine logische Bezeichnung zuteilen. Die COM-Schnittstelle soll in weiterer Folge eine wichtige Rolle spielen, da diese für die Eingabe „*Connect to COM*“ in Unity 3D benötigt wird um eine Verbindung zu den einzelnen Schnittstellen aufbauen zu können.

Im Allgemeinen sind serielle Schnittstellen, die grundsätzlich zur Datenübertragung zwischen zwei Geräten dienen, auch heute noch im industriellen Umfeld häufig anzutreffen und sind für einfache technische Anwendungen noch immer vollkommen ausreichend. RS-232 (*Recommended Standard* 232) ist beispielsweise ein Standard für eine serielle Schnittstelle, die in den 2010er Jahren und auch bereits davor bei Rechnern eine häufig verwendete Schnittstelle war.

Abschließend sollen Messwerte (relative Luftfeuchtigkeit und Temperatur) über Modbus TCP in die virtuelle Welt übertragen und über eine zusätzliche angebundene Applikation visualisiert werden.

5.1 Blackbox-Modellerstellung eines Digital Twins

In den folgenden Absätzen werden erste Überlegungen sowie die erste theoretische Vorauswahl, also das erste Grobkonzept (*Blackbox*-Modell) für die Umsetzung der Anforderungen in Hard- und Software im Überblick skizziert. Ein Modell ist grundsätzlich ein stark vereinfachtes Abbild der Realität, soll einen ersten Entwurf maßstäblich darstellen und einer weiteren Analyse im Zuge der Umsetzung dienen.

Ein einfaches, aber übersichtliches *Blackbox*-Modell soll dabei in einer theoretischen Vorauswahl (Konzept) einerseits das wesentliche Sollverhalten und die erwartete Interaktion des Systems mit seinen internen und externen Schnittstellen, als auch andererseits das daraus abzuleitende Umsetzungslayout in Hard- und Software zeigen.

Im Laufe der Softwareentwicklung soll, auf Basis dieses Blackbox-Modelles, entschieden werden ob noch zweckmäßige Adaptierungen, Modifizierungen und Optimierungen in Hard- und Software durchzuführen sind, die dann ebenfalls in den nächsten Kapiteln und Unterkapiteln mit einfließen. Zum leichteren Verständnis sollen auch die wesentlichsten Auswahlkriterien bzw. Informationen zur Umsetzung zu den einzelnen eingesetzten und erstellten Hard- und Softwarekomponenten mit einfließen. Diese sollen auch dazu dienen die Überlegungen zur Umsetzung des Konzeptes und der einzelnen Kriterien nachvollziehen zu können. Die fachlichen Termini, aus diversen Literaturquellen, werden für die leichtere Lesbarkeit, im Glossar näher erläutert.

Die relative Luftfeuchtigkeit und die aktuelle Temperatur im Raum sollen über einen einfachen, kompakten, kleinen aber leistungsfähigen DHT11 Sensor gemessen werden, der an einem Arduino kompatiblen Board angeschlossen wird und die somit ermittelten Werte sollen an einen (Dell-Alienware)-Rechner gesendet werden. Der digitale Sensor DHT11 soll dabei die relative Luftfeuchtigkeit in einem Messbereich von ca. 20 bis 80 % und die Temperatur in einem Messbereich von ca. 0 bis 50 °C messen.[39]

Die Temperatur und die Luftfeuchtigkeit sollen sowohl digital über ein Display in der Virtuellen Welt, als auch in Form von analogen Werten über ein zusätzlich modelliertes und virtuell dargestelltes Thermometer bzw. ein virtuell angezeigtes Hygrometer visualisiert werden.

Ein großes am virtuellen Messtisch über Unity 3D eingefügtes Display, das für die Darstellung von 3 Vergleichswerten programmiert werden soll und das die real ermittelten Messwerte auch digital anzeigt, soll nicht nur der optischen Darstellung dienen, sondern auch die Veränderung der Messwerte über einen frei auswählbaren Zeitraum in der virtuellen Welt bewusst sichtbar machen.

Neben der seriellen Schnittstelle sollen die gemessenen Messwerte der relativen Luftfeuchtigkeit und der aktuellen Temperatur über eine zusätzliche Applikation mittels TCP in der virtuellen Welt angezeigt werden.

5.2 Umsetzungslayout eines Digital Twin-Modelles

Auf Grund der bisherigen Anforderungen, Kriterien und Überlegungen soll dazu ein erstes einfaches Blackbox-Modell entworfen werden. Dieser erste Entwurf bzw. die erste grobe Skizze ist in der

[39] Vgl. Snieders (2019), Online-Quelle [01.05.2019].

nachfolgenden Abbildung (siehe dazu Abb. 24) dargestellt und erhebt in keinster Weise einen Anspruch auf Vollständigkeit und gutes Design und lässt zudem noch Freiraum für Anpassungen bzw. Veränderungen, die sich im Laufe der Umsetzung ergeben, offen.

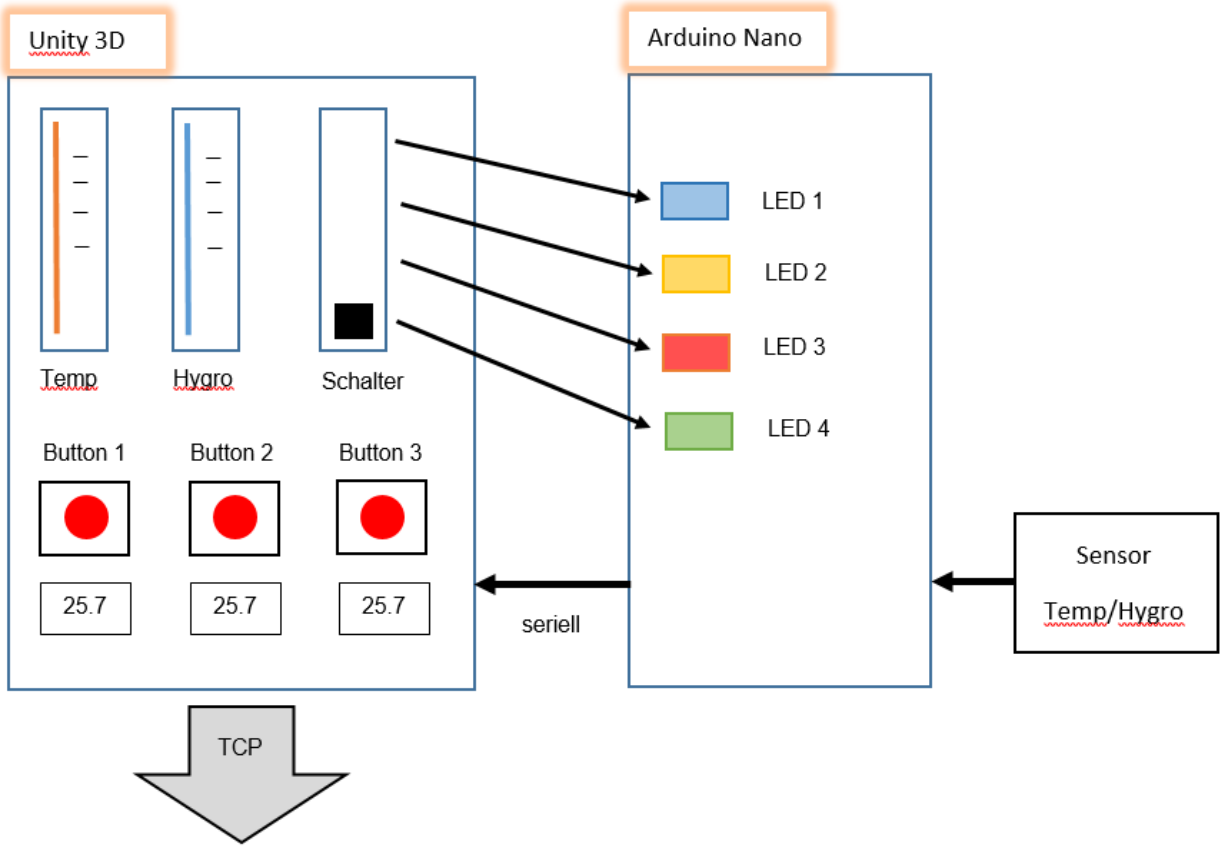


Abb. 24: Grober Entwurf eines Blackbox-Modelles und Groblayout für den Entwurf in Hard- und Software, Quelle: Eigene Darstellung.

Dieser einfach designte virtuelle Messtisch mit seinen Schnittstellen soll in weiterer Folge als Dummy dienen, um durch einen real existierenden, in digitaler Form abgebildeten virtuellen Zwilling im Elektro-Energietechniklabor der FH CAMPUS 02, nach Fertigstellung dieser Masterarbeit ersetzt zu werden.

Daher wird der Schwerpunkt auch auf das Schaffen und die Interaktion zwischen den einzelnen Schnittstellen gelegt und es ist eine Zielsetzung auf ansprechende Designs in der Entwicklungsumgebung bewusst zu verzichten.

Das Ersetzen dieses Dummys durch reale, virtuell dargestellte Objekte aus dem bestehenden Labor der FH CAMPUS 02 wird im Rahmen dieser Masterarbeit nicht umgesetzt und dieses Thema wird daher auch nur sehr peripher gestreift.

6 ARDUINO NANO-BOARD V3.0

Der Name Arduino™ bezeichnet grundsätzlich sowohl ein einfaches, kleines und günstiges Mikrocontroller-Board als auch die zugehörige Programmiersprache und das Entwicklungswerkzeug. Die über Stiftleisten nach unten herausgeführten Pins passen auf nahezu alle gängigen Stecksysteme. Die Rechte für die Marke „Arduino“ gehören der Firma tinker.it. Beim Nano handelt es sich um einen besonders kleinen Arduino, der speziell für die Arbeit mit Steckboards entwickelt wurde, der aber trotz der kleinen Baugröße über einen sehr leistungsstarken Mikrocontroller vom Typ ATmega 328/P verfügt.

Das Konzept dahinter ist es, dass die Entwicklung von Anwendungen für dieses leistungsstarke Mikrocontroller-System auf praktischem Weg durch Aufbauen und Testen erfolgt. Eine Lösung soll schnell und effizient als Prototyp aufgebaut werden. Im Anschluss daran kann man überprüfen, ob das Ergebnis auch der ursprünglichen Idee entspricht und ob die Anwendung auch das gewünschte und erwartete Verhalten aufweist.

Auf Grund dieser umfänglich angeführten Vorzüge und da neue Programme auch sehr einfach auf den leistungsstarken Mikrocontroller übertragen werden können, wurde, wie bereits in Kapitel 5 analysiert, für die Umsetzung des praktischen Teiles dieser Masterarbeit ein Arduino kompatibles Nano Board V3.0 ausgewählt.

6.1 Arduino Hardwarekomponenten

Die Arduino-Plattform besteht grundsätzlich aus einem Hardware- und einem Softwareteil. Die Hardware des Arduino-Projektes ist eine Leiterplatte, die mit verschiedenen Elektronik-Bauteilen bestückt ist. In der Praxis wird diese Leiterplatte auch als *Board* bezeichnet.

Die kostenlos verfügbare Entwicklungsumgebung, auch *Arduino Integrated Development Environment* (Arduino IDE) genannt, stellt den Softwareteil dar. Die Datenübertragung erfolgt seriell, also Bit für Bit (d.h. bitseriell).

Die physische Verbindung zum Hardwareteil muss während der Entwicklungs- und Testphase immer sichergestellt und aufrechterhalten bleiben. Nach der Programmerstellung kann die Verbindung zur Entwicklungsumgebung getrennt werden, und der Arduino kann dann als unabhängiger „Minicomputer“ mit der Außenwelt bzw. mit seinen Schnittstellen kommunizieren. Sollen Daten zur Weiterverarbeitung oder zur Anzeige an den Computer gesendet werden, muss die serielle Verbindung über den USB-Anschluss auch immer physisch aufrecht erhalten bleiben.[40]

Das in der nachfolgenden Abb. 25 dargestellte Arduino Nano-Board ist, wie schon eingangs erwähnt, eine einfache und kostengünstige Leiterplattenversion von geringer Baugröße, die dennoch nahezu den Funktionsumfang des UNOs vorzuweisen vermag. Die Abmessung der Platine beträgt lediglich (Breite x Länge) 18,5 mm × 43 mm.

[40] Vgl. Brühlmann (2015), S. 19 ff.

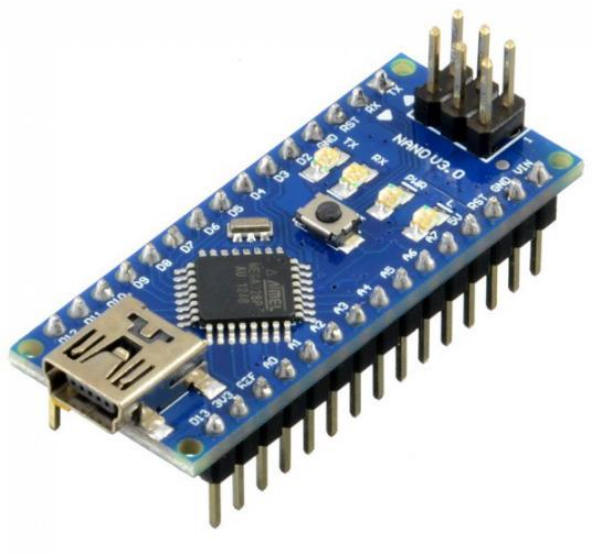


Abb. 25: Arduino Nano-Board V3.0, Quelle: core Electronics (2019), Online-Quelle [01.05.2019].

Auf dem Arduino Board stehen alle notwendigen Komponenten inklusive USB-Schnittstelle und Spannungsregler zur Verfügung. Das Arduino kompatible Nano-Board besitzt annähernd dieselbe Funktionalität wie ein jedes andere Standard-Board, aber anstelle von sechs besitzt es sogar acht analoge Ports. Durch den Aufbau der beiden Stiftreihen am Rand kann das Arduino Nano-Board ideal beim Testen der Entwicklung auch auf Steckbrettern (Steckboards oder *Breadboards*) eingesetzt werden und passt nahezu auch auf alle anderen gängigen Stecksysteme.[41]

Über den Mini-USB-Anschluss können sowohl *Board* und Schaltung mit Strom versorgt und auch Programme auf den Mikrocontroller übertragen werden. Derzeit ist bereits die aktuelle Version V3.3 verfügbar.

6.2 Pinbelegung und Pinbeschreibung des Arduino Nano Boards

Die Pinbelegung sowie die Pinbeschreibung (siehe dazu auch Abb. 26 bis 29) des Arduino Nano-Boards sind für die unterschiedlichen Anschlussmöglichkeiten und für die weitere Programmerstellung und -ausführung maßgeblich.

Da es sich beim in Abb. 30 sowie in Abb. 38 dargestellten Sensor DHT11 allerdings um einen digitalen Sensor handelt, ist in Widerspruch zur Bilddarstellung eine Korrektur der Pinbelegung notwendig, um Messwerte zu ermitteln und übertragen zu können.

In der Abb. 26 ist daher klar zu erkennen, dass der Sensor DHT11 mit dem richtigen Eingang verbunden werden muss um eine einwandfreie Funktionalität herzustellen. Beim DHT11 handelt es sich, wie bereits mehrmals angeführt um einen digitalen Sensor, der die relative Luftfeuchtigkeit und die Raumtemperatur misst.

[41] Vgl. Brühlmann (2015), S. 50.

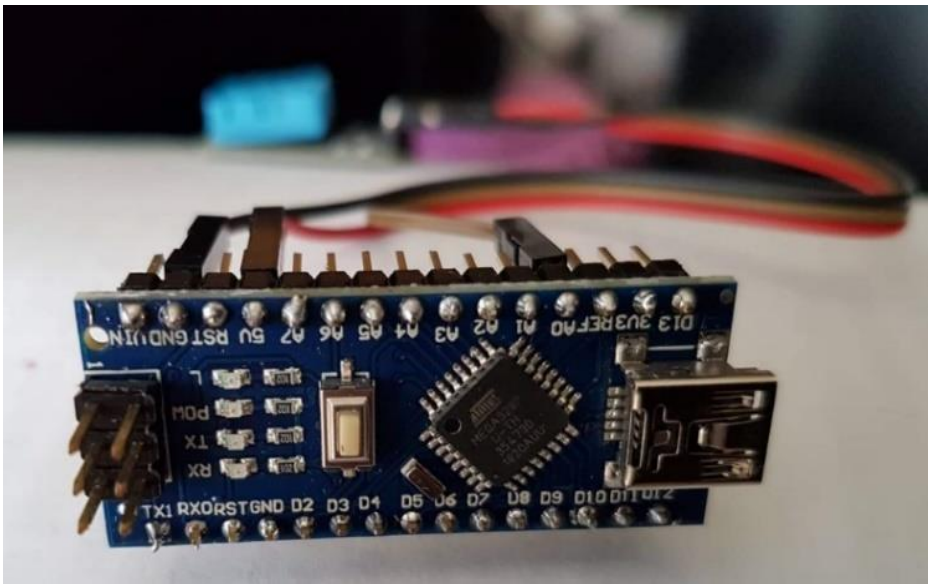


Abb. 26: Pinbelegung des Arduino Nano-Board V3.0 für den DHT11 Sensor, Quelle: Eigene Darstellung.

Die folgenden 3 Abbildungen zeigen einen Überblick über die Pinbelegung bzw. -beschreibung des Arduino kompatiblen Nano-Boards aus unterschiedlichen Blick- und Betrachtungswinkeln. Die Abbildungen stammen von unterschiedlichen Herstellern.

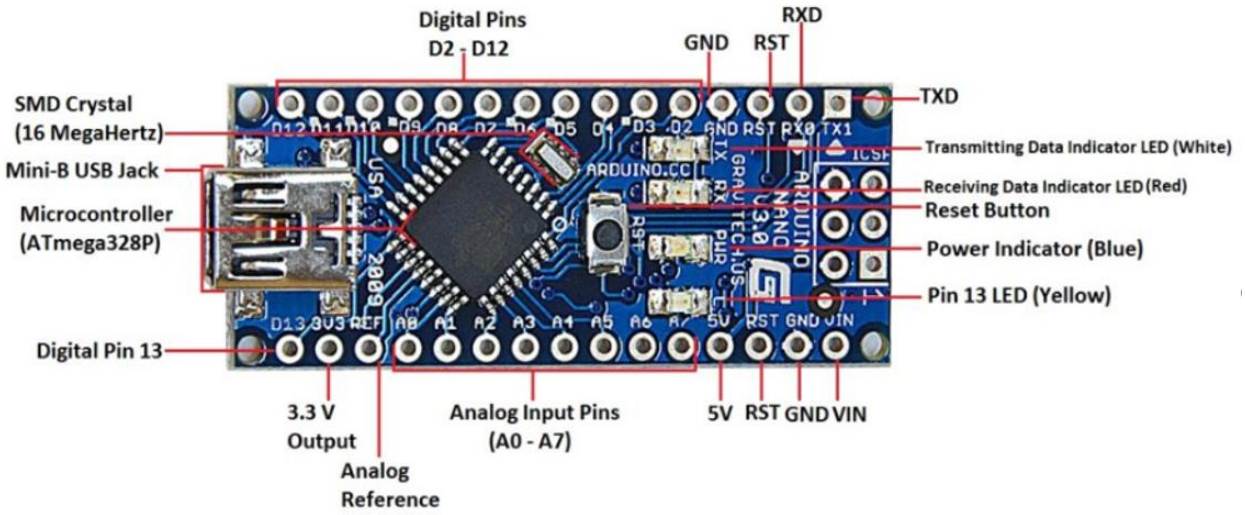


Abb. 27: Pinbelegung des Arduino Nano-Board V3.0, Quelle: CIRCUITS TODAY (2018), Online-Quelle [01.05.2019].

Die digitalen Ein- und Ausgänge D0 (RX0) und D1 (TX1) dienen rein der Datenübertragung zwischen dem Mikrocontroller der Type ATmega 328/P und dem Computer. Diese Pins dürfen daher nicht mit anderen Funktionen belegt werden, um die Kommunikation zwischen dem Mikrocontroller und dem Computer nicht zu stören.

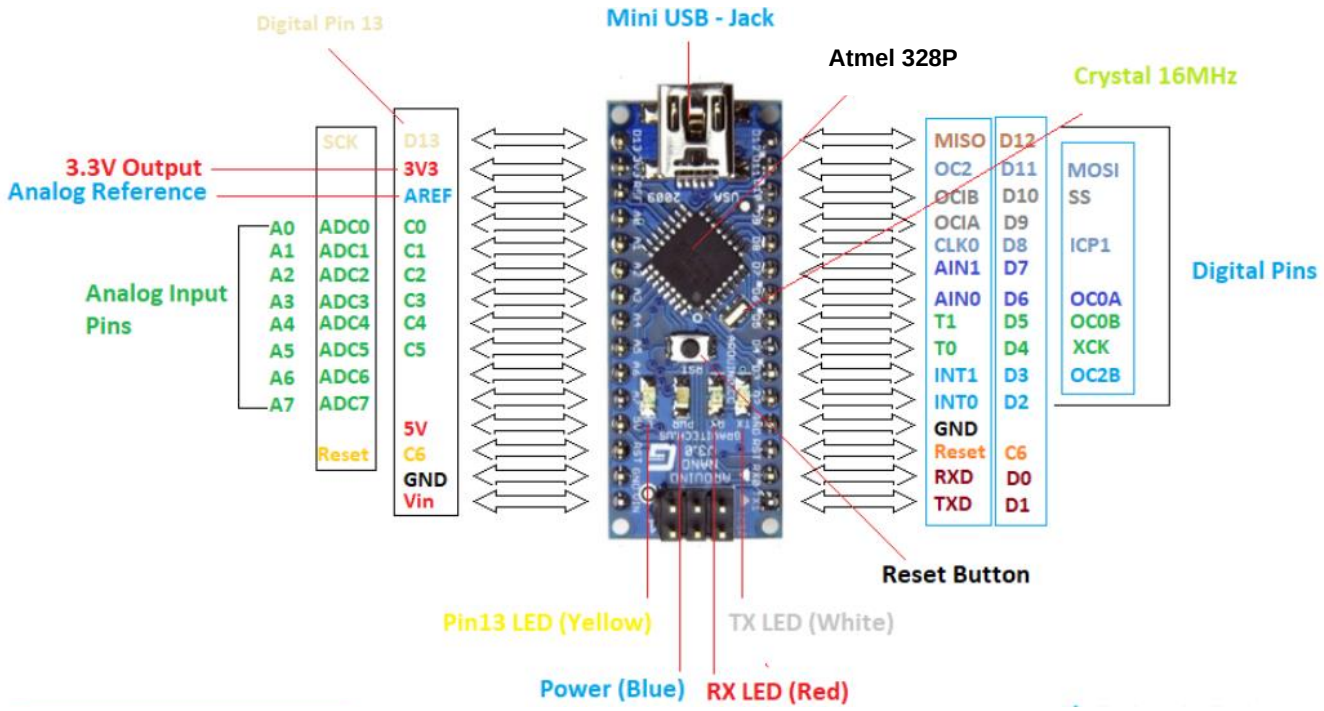


Abb. 28: Pinbeschreibung des Arduino Nano-Board V3.0, Quelle: The engineering projects (2018), Online-Quelle [01.05.2019].

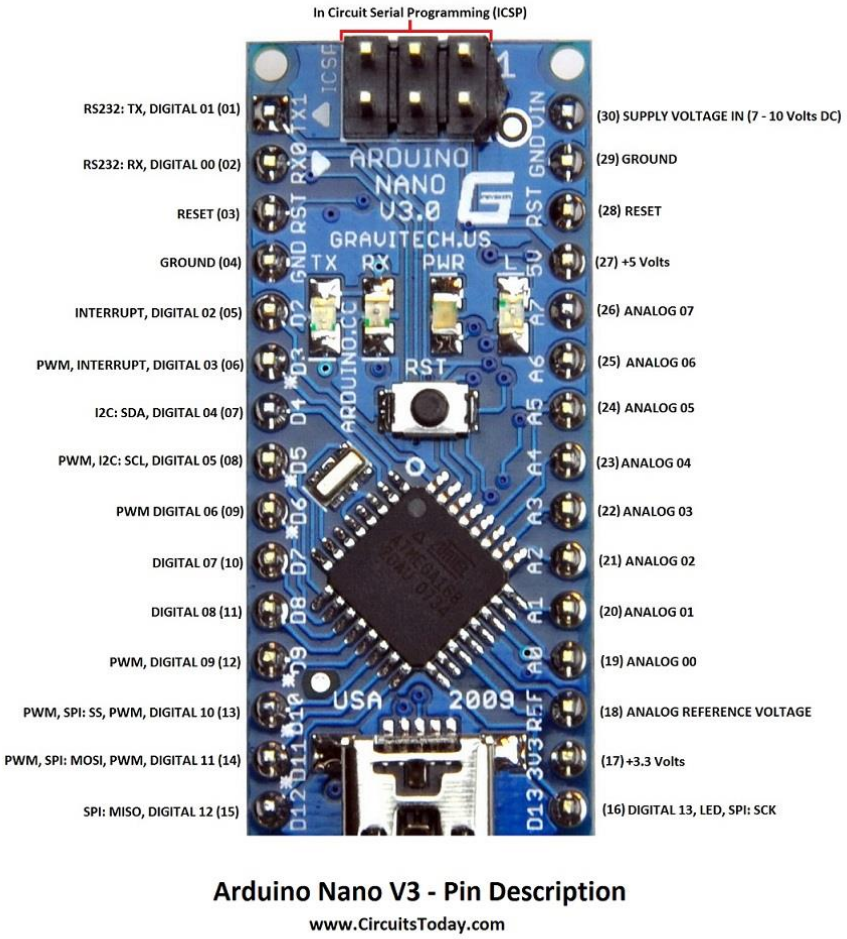


Abb. 29: Pinbeschreibung des Arduino Nano-Board V3.0, Quelle: CIRCUITS TODAY (2018), Online-Quelle [01.05.2019].

Anschlussmöglichkeiten, Bauteile bzw. Funktionen

Das verwendete Arduino kompatible Nano-Board hat folgende wesentliche Anschlussmöglichkeiten und Schnittstellen, Bauteile bzw. weist folgende Funktionen auf:[42]

Anschlüsse/Schnittstellen:

Mini-B-USB, SPI, 1 x I²C, USART, ICSP (für die Programmierung ohne *Bootloader*)

Externe Stromversorgung:

- Pin 30: Empfohlene Eingangsspannung zwischen 7 bis 12 V (ungeregelte externe Stromversorgung). Der Grenzbereich der Eingangsspannung liegt zwischen 6 bis 20 V.
- Pin 27: Geregelte 5 V Spannungsversorgung (Betriebsspannung, logische Ebene).
- Pin 13: Konstante 3,3 V-Versorgung (FTDI output) durch integrierten Spannungsregler, die maximale Stromaufnahme für diesen 3,3 V-Pin beträgt 50 mA.

Gewicht des bestückten Arduino Nano-Boards V3.0:

Das Gewicht beträgt ca. 7 g.

I/O-Pins (digitale Pins von D0 bis D13):

Jeder dieser Anschlüsse kann als Eingang für Schalter oder als Ausgang, z.B. für LEDs (Leuchtdioden) benutzt werden. Der maximale Strom pro I/O-Pin beträgt 40 mA. Das reicht um eine LED mit passendem Vorwiderstand hell leuchten zu lassen. Einige dieser Pins erfüllen z.B. noch Sonderfunktionen:

- Pulsweitenmodulation (PWM)
- Analog (A0 bis A7)
- Digital (D0 bis D13)
- Serielle Pins: RX und TX, Pin 0 (RX0) für das Empfangen und 1 (TX1) für das Senden von seriellen TTL-Daten.

Leuchtdioden am Arduino Board:

- ON (*Power*): Diese LED leuchtet (Leuchtfarbe am Arduino Board: blau) sobald das Arduino Nano-Board mit Strom versorgt wird.
- TX1 und RX0: Diese dienen zur Visualisierung der seriellen Kommunikation des Boards. Wenn das Arduino Nano-Board programmiert wird, blinkt die RX0-LED (Leuchtfarbe am Arduino Board: rot), die das Ankommen von Daten anzeigt.
  Die TX1-LED (Leuchtfarbe am Arduino Board: weiß) zeigt an, wenn Daten vom Controller gesendet oder von Peripheriegeräten empfangen werden.
- L: Die LED L (Pin 13, Leuchtfarbe am Arduino Bord: gelb) kann direkt angesteuert und getestet werden, ohne die externe Hardware anschließen zu müssen.

[42] Vgl. Atmel ATmega328P-Datenblatt (2015), Online-Quelle [01.05.2019].

Mikrocontroller:

Der Mikrocontroller führt die Programme aus und verarbeitet die Ein- und Ausgangssignale (z.B. Messwerte wie Temperatur, Luftfeuchtigkeit …). Beim Arduino Nano kommt standardmäßig ein Atmel ATmega 328/P zum Einsatz.

Die wichtigsten Eckdaten des Mikrocontrollers sind:

Speicher:

- 32 kB Flash-Speicher (2 kB davon werden für den Bootloader benötigt). Der Bootloader ist ein kleines Programm, der für das Laden des eigentlichen Programmes verantwortlich ist.
    - 1 kB EEPROM
    - 2 kB RAM

CPU:

Atmel AVR CMOS-8-Bit Mikrocontroller, der auf der AVR-RISC-Architektur basiert.

Weitere Features:

- 16 (Standard) – 20 MHz maximale Clock-Frequenz (unterschiedlich je Hersteller)
- Integrierter USB-Controller
- 2 externe Interrupt-Pins (Pin 2 und 3), die zum Aktivieren eines Interrupts verwendet werden

Ein- und Ausgänge:

- 8 analoge Eingänge (ADCs), A0 bis A7 und
- 14 digitale Input/Output-Pins, D0 bis D13, wovon 6 als PWM Kanäle nutzbar sind (z.B. Pins 3, 5, 6, 9, 10 und 11)

Zulässige Arbeitsspannung:

- 1,8 V bis 5,5 V

6 Sleep Modes:

- *Idle*,
- *ADC noise reduction,*
- *Power-save,*
- *Power-down,*
- *Standby* und *extended standby.*

Mini-B-USB-Anschluss:

Über den USB-Anschluss wird das Arduino Board mit Strom und den aktuellsten Programmen versorgt. Er liefert 5 V und bis zu 500 mA für das Board und die angeschlossene Peripherie.

Es ist also möglich mehrere LEDs anzuschließen, selbst wenn keine externe Stromversorgung zusätzlichen Strom liefert.

Power-Pins:

Power-Pins sind keine I/O-Pins. Es handelt sich hierbei um Anschlüsse, an denen Spannung abgegriffen oder zugeführt werden kann.

- GND:

  Das Arduino Board verfügt über 3 *Ground-Pins*. Diese sind Masse-Pins um z.B. Leuchtdioden anzuschließen oder einen Eingang auf *Low* (Masse-Niveau) zu schalten. GND ist der Erdungsstift der Platine.

- 5 V-Ausgang:
  Ein konstanter Ausgang von 5 V für die externe Hardware und einer maximalen Belastbarkeit von 100 mA. Geregelte Versorgungsspannung, die zur Versorgung der Platine sowie der Komponenten dient.
- 3,3 V-Ausgang:

  Konstanter Ausgang von 3,3 V, maximale Belastbarkeit: 50 mA. 3,3 V ist die Mindestspannung, die vom Spannungsregler auf der Platine erzeugt wird.

- Vin:

  Vin ist die Eingangsspannung der Platine. Anschlussmöglichkeit für eine externe Stromquelle (z.B. eine Batterie) mit 9 bis 12 V als Spannungsversorgung.

Reset-Taster (*Button*):

Durch Drücken des *Reset*-Tasters erfolgt ein Neustart und Zurücksetzen des Mikrocontrollers. Eingespielte Programme bleiben dabei erhalten, werden aber an den Anfang zurückgesetzt.

SMD Schwingquarz:

Quarzoszillator ist der Taktgeber des Controllers. Er stellt die Taktfrequenz zwischen 16 MHz (im Standard) und 20 MHz (unterschiedlich je Hersteller) für den Mikrocontroller bereit.

Stecker für ICSP:

Stecker für *In-Circuit Serial Programming*, Stiftleiste 2 x 3-polig; Mit diesem Anschluss und einem externen Programmiergerät (z.B. dem STK500) ist es für Anwender, die im Umgang mit Mikrocontrollern erfahren sind auch möglich, den Arduino direkt zu programmieren oder den internen *Bootloader* zu überschreiben.

Steckerleisten:

Es stehen 8 analoge Eingänge zur Verfügung. Das bedeutet, es gibt hierbei die analogen Eingänge A0 bis A7, im Spannungsbereich von 0 bis 5 V und 14 digitale Ein- und Ausgänge D0 bis D13, im Bereich von 0 V (low) bis 5 V (high) sowie jene die mit den Funktionen *using pinMode()*, *digitalWrite()*, und *digitalRead()*, für den Anwender ausgestattet sind. [43]

[43] Vgl. Kainka (2013), S. 23 ff.

7 DIGITALER TEMPERATUR- UND LUFTFEUCHTIGKEITS-SENSOR DHT11

Für das Einlesen der Temperatur- und Luftfeuchtigkeit wird ein kostengünstiger Sensor, vom Typ DHT11 (Abmessung der Platine: ca. 31 mm x 14 mm) eingesetzt. Dieser 3-Pin-Sensor von Aosong Electronics verfügt über ein eigenes Kommunikationsprotokoll (*OneWire*-Kommunikation).[44]

Er wird direkt über das LED-Board mit dem Arduino kompatiblen Board verbunden. Wie bereits in Unterkapitel 6.2 ausgeführt ist auch hier auf die richtige Pinbelegung zu achten.

Der Vorteil dieses Sensors ist es, dass in einer kompakten Bauform eine Kombination aus Temperatur- und Luftfeuchtigkeitsmessung geschaffen wird und dass der Sensor sich gut für Langzeit-messungen eignet. Der Nachteil ist allerdings seine geringe Abtastrate, sodass nur alle 2 Sekunden ein neuer Messwert zur Verfügung steht.

Der Sensor DHT11, wie in der Abb. 30 dargestellt, sitzt in einem Gehäuse auf einer Platine, der über einen 3 Pin-Stiftsockel (GND: Masse, DATA: Datenleitung und VCC: Versorgungsspannung) auf einen Stecker über die Anschlusskabel (*Jumpwires*) geführt wird.

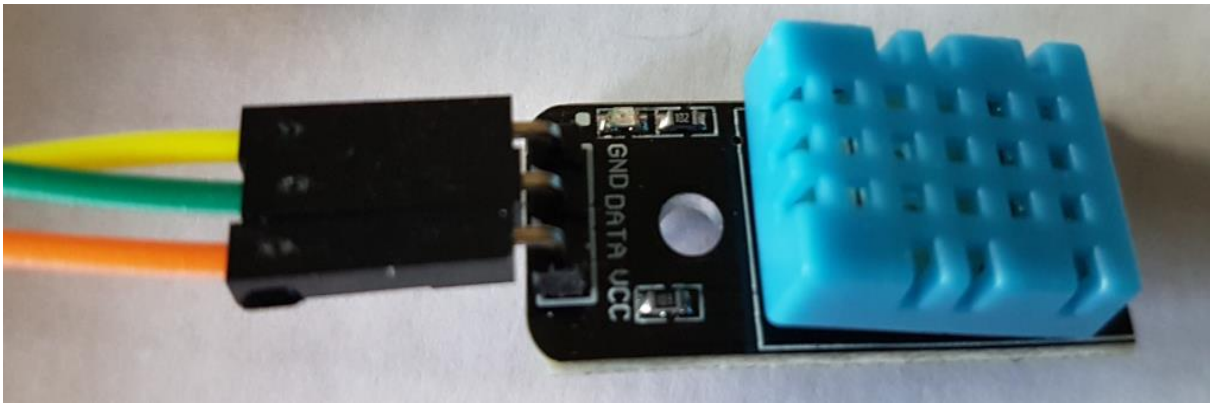


Abb. 30: Digitaler Temperatur- und Luftfeuchtigkeitssensor DHT11, Quelle: Eigene Darstellung.

Der Sensor verfügt über einen kapazitiven Feuchtigkeitssensor sowie einen Thermistor und ermittelt die Messwerte mit folgender Genauigkeit:

- Relative Luftfeuchtigkeit: zwischen ca. 20 bis 80 % (mit einer Genauigkeit von 5 %) und
- Temperatur: zwischen ca. 0 bis 50 °C (mit einer Genauigkeit von +/- 2 °C).[45]

Neue Daten (Messwertänderungen) werden, wie schon eingangs angeführt, lediglich mit einer Abtastrate von ca. 2 sec (nichtdeterministisch) abgerufen und übertragen.

Diese Messgenauigkeit und die Latenzzeit sind für das Elektro-Energietechnik-Labor der FH CAMPUS 02 grundsätzlich ausreichend und liefern daher für diese Masterarbeit durchaus angemessene Werte.

[44] Vgl. Kappel (2016), S. 211.

[45] Vgl. Snieders (2019), Online-Quelle [01.05.2019].

In weiterer Folge werden diese Messwerte über die serielle Schnittstelle in die Spielesoftware Unity 3D 2018 übertragen und danach visualisiert und auch über eine weitere Schnittstelle, Modbus-TCP, virtuell angezeigt. Über Modbus-TCP können die ermittelten Daten anschließend einer weiteren Verwendung, z.B. zur Erstellung von grafischen Auswertungen zugeführt werden.

7.1 Die Datenübertragung

Um den Sensor richtig einsetzen zu können, muss man sich eingangs näher mit diesem vertraut machen und die Art der Datenübertragung verstehen.

„Der digitale Sensor DHT11 basiert auf einer eigenen *OneWire*-Kommunikation. Dabei werden über die Spannungsversorgung Daten übertragen, indem diese einfach kurzzeitig auf 0 V gezogen wird. Die Peripheriegeräte müssen über Kondensatoren verfügen, um diese Zeit ohne Spannungsversorgung zu überbrücken.

Aus denselben Gründen wie bei I²C benötigt die Datenleitung einen *Pull-up*-Widerstand. Wenn der Arduino keine Daten sendet, so hat die Leitung ein hohes Potenzial (d.h. die Übertragungsleitung liegt im Ruhezustand auf dem *High*-Pegel). Dadurch kann der Sensor den Pegel wieder nach unten ziehen, um etwas zu signalisieren.

Wenn der Arduino Daten vom Sensor lesen will, so zieht er die Leitung für mindestens 18 ms auf den logischen Low-Pegel. In dieser Zeit bereitet der Sensor die Daten vor, danach folgt ein *High*-Pegel, der zwischen 20 und 40 µs lang sein sollte. Damit gibt man dem Sensor die Möglichkeit, eine Art Bestätigung zu senden.

Ist der *High*-Pegel durch den Arduino beendet, so kann der Sensor die Leitung auf *Low* ziehen, um zu signalisieren, dass er seine Aufgabe verstanden hat. Dies dauert 80 µs. Da die Daten auch immer auf einem *Low*-Pegel beginnen, muss die Leitung vorher noch einmal für 80 µs auf *High* gehen.

Vor jedem Datenbit wird der Pegel auf 0 gezogen, und zwar für 50 µs. Danach folgt das Datenbit. Anhand der Länge des Pulses kann man herauslesen, ob es eine logische 0 (wenn der Puls zwischen 26 und 28 µs lang war) oder eine logische 1 (wenn der Puls länger als 50 µs lang war) gewesen ist.

Sind keine Daten mehr zu übertragen, geht die Leitung durch den *Pull-up*-Widerstand wieder auf *High*, und die Übertragung kann erneut erfolgen.

Insgesamt werden 40 Bits (d.h. 5 Bytes) (siehe dazu auch Abb. 31) wie folgt übertragen:

- Auflösung für die Luftfeuchtigkeit: 16 Bits,
- Auflösung für die Temperatur: 16 Bits und
- Prüfsumme: 8 Bits.

Die ersten beiden Bytes dienen für die relative Luftfeuchtigkeit, das dritte und vierte dienen für die Temperatur, und das fünfte ist eine Prüfsumme, um zu überprüfen, ob die Übertragung fehlerfrei war (siehe dazu auch das Berechnungsbeispiel in Unterkapitel 7.2).

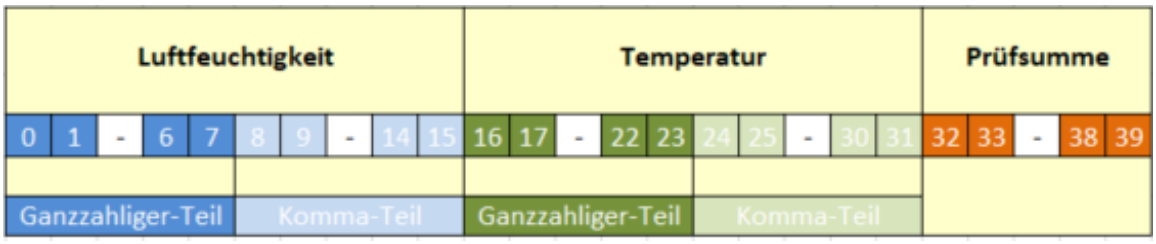


Abb. 31: Datenformat der Übertragung des Sensors DHT11, Quelle: Eigene Darstellung.

Die Prüfsumme dieses Sensors wird gebildet, indem alle übertragenen Bytes außer dem Byte für die Prüfsumme addiert werden. Die Prüfsumme entspricht dann den niederwertigsten Bits dieser Addition. Die Daten wurden korrekt übertragen, wenn die übertragene Prüfsumme der aus den empfangenen Daten berechneten Prüfsumme entspricht."[46]

7.2 Berechnungsbeispiel

Über den digitalen Sensor DHT11 werden testweise, die nachfolgenden in Abb. 32 dargestellten Messwerte ermittelt. Es werden dabei 40 Bits übertragen und daraus resultierend die folgenden Daten bzw. Werte empfangen:

Abb. 32: Berechnungsbeispiel für 40 über den Sensor erhaltene Bit-Daten, Quelle: Eigene Darstellung.

Durchführung der Berechnung:

$0011\ 1100_2 + 0000\ 0000_2 + 0001\ 0111_2 + 0000\ 0000_2 = 0101\ 0011_2$

Vergleich der Quersumme mit der Prüfsumme: „Die empfangenen Daten sind somit korrekt."

- Relative Luftfeuchtigkeit:

  $0011\ 1100_2 = 3C_{16} = 60_{10} = 60\%$, Ergebnis: Wert ist richtig

- Temperatur:

  $0001\ 0111_2 = 17_{16} = 23_{10} = 23\,°C$, Ergebnis: Wert ist richtig

[46] Vgl. Kappel (2016), S. 211 – 214.

8 DIE LED-LEITERPLATTE (PROTOYP EINES LED-BOARDS)

Für die Ansteuerung von lichterzeugende Dioden (LEDs) über die virtuelle Welt und die Anzeige in der realen Welt soll anstelle der Verwendung eines klassischen Steckbrettes (engl. *Breadboard*) eine selbst gefertigte Leiterplatte (LED-Board) als Prototyp verwendet werden, der als *Shield* (aufsteckbare Zusatzplatine) direkt auf die Pins des Arduino kompatiblen Boards aufgesetzt und somit mit dem Arduino Board kontaktiert wird um die Daten zu übertragen.

Das Arduino-kompatible Board ist zwar mit 4 LEDs bestückt, diese sind aber als Power-LED (Leuchtfarbe: blau) (siehe dazu auch Unterkapitel 6.3), und als Anzeige für das Empfangen (Leuchtfarbe: rot) und Übertragen von Daten (Leuchtfarbe: weiß) bereits belegt und lediglich 1 LED (Leuchtfarbe: gelb) kann frei über den Anwender programmiert und angesteuert werden (siehe auch Unterkapitel 6.2)

8.1 Das Leiterplattendesign

Um vier zusätzliche Leuchtdioden (LEDs) über das Arduino kompatible Board ansteuern zu können, besteht entweder die Möglichkeit, für den elektronischen Schaltungsaufbau ein *Breadboard* mit den notwendigen Bauteilen und Drähte (LEDs in unterschiedlichen Leuchtfarben, Drähte (*Jumper*), Widerstände, etc.) einzusetzen oder selbst für diese Anwendung einen einfachen Prototyp zu entwerfen und zu bestücken. Da die Verwendung eines *Breadboards* mit den Bauteilen keine ausreichend gute mobile Anwendung unterstützt und die Flexibilität eingeschränkt ist, ist es effizienter und auch umsetzungstechnisch wesentlich interessanter ein LED-Board selbst zu entwerfen und danach dann herstellen zu lassen.

Dieses LED-Board soll als kostengünstiger Prototyp dienen um Tests zur Ansteuerung der LEDs über die serielle Schnittstelle in die virtuelle Welt und umgekehrt aus der virtuellen Welt in die reale Welt durchzuführen und die Funktionalität der Schaltung zu visualisieren. Das Leiterplattenlayout (siehe dazu auch Abb. 34) soll in der Entwurfssoftware für Leiterplatten-Layouts und Stromlaufpläne „Eagle Standard“ erstellt und in ein dxf-file zur Darstellung in „Solid Works eDrawings 2018 x64 Edition“ konvertiert werden.

Abb. 33: EAGLE Logo, Quelle: AUTODESK (2019), Online-Quelle [30.04.2019].

„Eagle Standard“ enthält grundsätzlich 99 Schaltplanblätter, 4 Signallagen und 160 cm² Platz für Leiterplatten. Eagle Standard verfügt zusätzlich über eine umfangreiche Bauteilbibliothek für die Erstellung des Platinenlayouts. Die Software selbst steht kostenlos auf der Homepage von AUTODESK, Inc. zum Download zur Verfügung. Die Aktivierung der Software (Abonnement) kostet jedoch 17,85 EUR / Monat oder 130,90 EUR / Jahr.

Die unbestückte Leiterplatte selbst wird anhand des erstellten Leiterplattenlayouts mit den gewünschten Leiterbahnen, Durchkontaktierungen (*Vias*), Bohrungslöchern für die Bauteile und mit den Pins über die Firma Conrad Electronic in Graz gefertigt und zugekauft (siehe dazu auch Abb. 34 und 35).

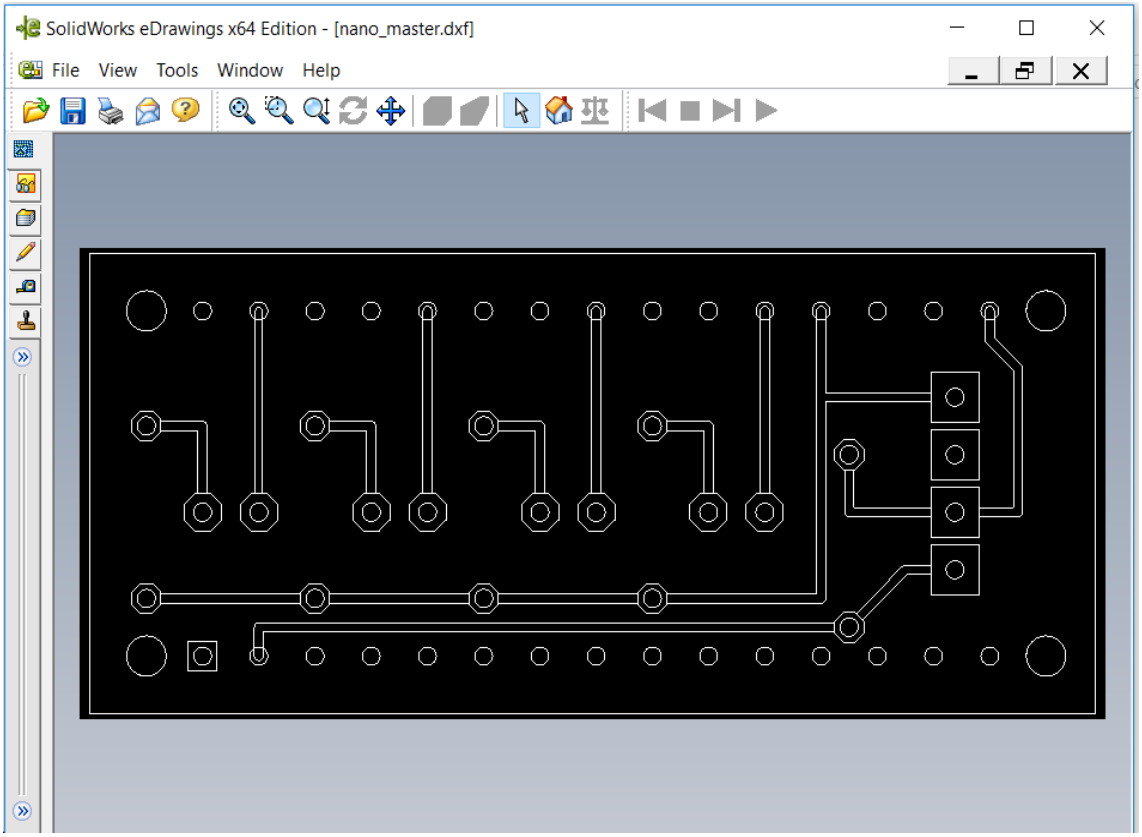


Abb. 34: LED-Board-Leiterplattenlayout, Quelle: Eigene Darstellung.

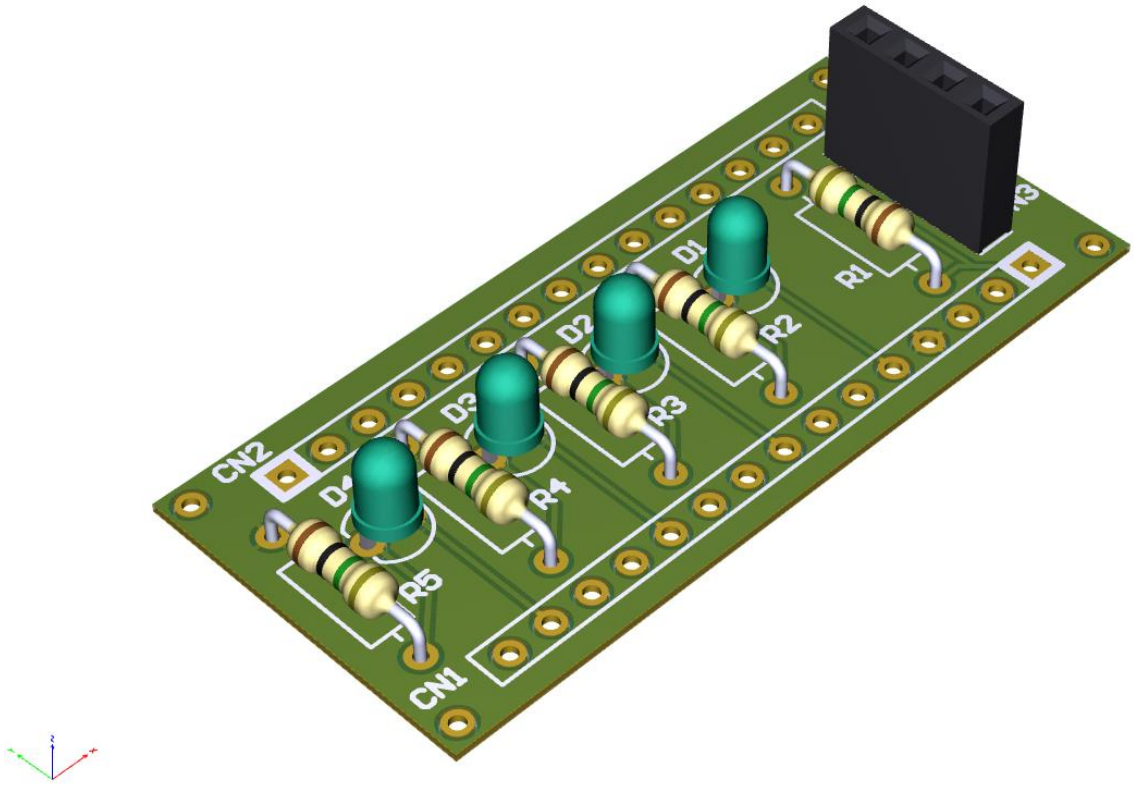


Abb. 35: LED-Board-3D-Leiterplattmodell, Quelle: FH CAMPUS 02, [18.11.2019].

8.2 Die Schaltungstechnik der Leuchtdioden (LEDs)

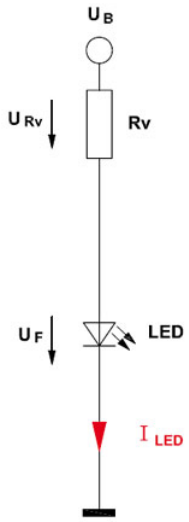


Abb. 36:LED-Schaltungstechnik Quelle: Elektronikwissen (2019), Online-Quelle [13.05.2019].

Die Leuchtdioden (LEDs) wandeln die elektrische Energie in Licht um und funktionieren wie Halbleiterdioden, die in der Durchlassrichtung Licht erzeugen (siehe dazu auch Abb. 36). Leuchtdioden reagieren sehr empfindlich auf einen zu großen Durchlassstrom (I_F). Deshalb darf eine Leuchtdiode niemals direkt an eine Spannungsquelle angeschlossen werden.

Eine Leuchtdiode muss immer mit einem Vorwiderstand (R_V) oder einem strombegrenzenden Bauteil beschaltet sein. Die Durchflussspannung (U_F), der Spannungsabfall im Betrieb, ist insbesondere von der Leuchtfarbe abhängig und ist dem technischen Datenblatt zu entnehmen.

Folgende Daten müssen für die Berechnung bekannt sein:

- Betriebsspannung (U_B) = 5 V
- Durchlassstrom (I_F) (gemäß technischem Datenblatt des Herstellers)
- Gewählter LED-Strom (I_{LED}), diesen kann man selber bestimmen
- Durchflussspannung (U_F) (gemäß technischem Datenblatt des Herstellers)

Beim Bestücken der Platine ist auch darauf zu achten, dass die LEDs polungsabhängig sind (Anode, Kathode).

Eine Leuchtdiode ist kein ohmscher Verbraucher, dessen Widerstand immer gleich ist, sondern ein Halbleiter, dessen Widerstand nach Anlegen einer Spannung gegen Null sinkt. Das bedeutet, der Strom steigt rein theoretisch unendlich an. Ein zu hoher Strom zerstört die Leuchtdiode.

Vor der Zerstörung tritt ein Temperaturanstieg ein. Die Leuchtdiode wird wärmer. Bekanntlich leiten warme Halbleiter besser als kalte. Es folgt also ein weiterer Stromanstieg, der dazu führt, dass die LED heiß und letztendlich zerstört wird.[47]

8.3 Bestimmen der Vorwiderstände (R_V) am LED-Board

Vorwiderstände sind grundsätzlich elektrische Widerstände und dienen dazu die elektrische Spannung bzw. die elektrische Stromstärke, die am Bauteil anliegt bzw. die durch das Bauteil fließt, zu begrenzen um dieses nicht zu zerstören.

Die Vorwiderstände für die Leuchtdioden (LEDs) (siehe dazu auch Kapitel 8.2) werden gemäß nachfolgender Formel berechnet:

$$RV = \frac{UB - UF}{ILED}$$

[47] Vgl. Schnabel (2017), Online-Quelle [13.05.2019].

Die technischen Daten der Leuchtdioden werden gemäß Datenblatt des Herstellers, der Firma JiangMen HuiYuan Opto-Electronic Co., Ltd. in China, in der nachfolgenden Tabelle übersichtlich dargestellt:

Kategorie / Bauart	Leuchtfarbe	U_F	I_F	Gehäuse	Lichtstärke	Wellenlänge
LED bedrahtet Bauart: THT	Grün	3,5 V	20 mA	5 mm	750 mcd	525 nm
LED bedrahtet Bauart: THT	Gelb	2,1 V	20 mA	5 mm	1400 mcd	588 nm
LED bedrahtet Bauart: THT	Klar Blau	3,2 V	20 mA	5 mm	2000 mcd	465 nm

Tabelle 7: Technische Daten der verwendeten LEDs, Quelle: Eigene Darstellung.

Da der Diodenstrom (I_F) gem. technischem Datenblatt des Herstellers bekannt ist, wird ein I_{LED} zwischen 10 bis 15 mA gewählt. Für das LED mit der gelben Farbe ergibt sich daher folgende Berechnung des Vorwiderstandes:

$$RVgelb = \frac{UB - UF}{ILED} = \frac{5\,V - 2{,}1\,V}{15\,mA} = 190\,\Omega$$

Auf Grund dieser Berechnung wird daher ein 220 Ω Widerstand ausgewählt.

Die eingesetzten Standard-Leuchtdioden haben einen Durchmesser von 5 mm. Sie zählen zu den am Häufigsten verwendeten Leuchtdioden in elektronischen Schaltungen. Sie beginnen schon bei 8 bis 12 mA zu leuchten. Erhöht man den Strom leuchten sie heller. Bei 20 mA ist die maximale Leuchtkraft erreicht. Der Unterschied zu 15 mA ist aber nur minimal. Meist ist ein Strom von ca. 10 mA schon ausreichend, um sie für die Anwendung zum Leuchten zu bringen.[48]

Für den Vorwiderstand der gelb leuchtenden LED wird daher ein 220 Ω Widerstand gewählt. Auch für die restlichen LEDs ist ein Vorwiderstand von ein 220 Ω ausreichend um sie zum Leuchten zu bringen und um sie vor einem zu hohen Stromdurchfluss zu schützen.

Die Platine ist im Anlieferzustand nicht fertig bestückt, sondern es müssen am LED-Board noch 4 Halbleiterdioden (4 Stück THT-LEDs) wie folgt aufgelötet werden:

- für die permanente Anzeige der aktuellen Raumtemperatur: 2 Stück LEDs klar in blauer Leuchtfarbe,
- für die Ansteuerung der LEDs in der virtuellen Welt: 1 Stück LED in grüner Leuchtfarbe und

[48] Vgl. Schnabel (2017), Online-Quelle [13.05.2019].

- für die Ansteuerung der LEDs in der virtuellen Welt: 1 Stück LED in gelber Leuchtfarbe sowie
- R2 – R5: 4 Stück 220 Ω Vorwiderstände und
- R1: 1 Stück 30 kΩ Pull-up-Widerstand (siehe dazu auch Unterkapitel 7.1)

8.4 Die Farbcodierungstabelle für bedrahtete Widerstände mit den 5 Farbringen

Um die verschiedenen bedrahteten Widerstände unterscheiden zu können sind diese, auf Grund der geringen Beschriftungsfläche, mit einer unterschiedlichen Anzahl an Farbringen gekennzeichnet (siehe dazu auch Abb. 37). Die für die Fertigung des LED-Boards verwendeten Widerstände sind Kohlewiderstände, mit flammhemmender Beschichtung und einer maximalen Verlustleistung von 0,25 W.

Abb. 37: Farbcodierung des Vorwiderstandes, Quelle: digikey (2019), Online-Quelle [24.05.2019].

Mit Hilfe von Farbcodierungstabellen wie die, in den zwei nachfolgenden Tabellen dargestellten, sollte der Widerstand bestimmt und vor der Bestückung der Platine nochmals kontrolliert werden:

Ring	Farbe	Korrespondierende Werte	Anmerkung
1	rot	2	
2	rot	2	
3	schwarz	0	
4 (Multiplikator)	schwarz	x 1 Ω	
5 (Toleranz)	braun	+/- 1 %	Axial
Widerstandswert	Vorwiderstand (R_V): 220 Ω +/- 1 %, 0,25 W, mit flammhemmender Beschichtung		

Tabelle 8: Farbcodierung der Vorwiderstände, Quelle: Eigene Darstellung.

Ring	Farbe	Korrespondierende Werte	Anmerkung
1	orange	3	
2	schwarz	0	
3	schwarz	0	
4 (Multiplikator)	rot	x 100 Ω	
5 (Toleranz)	braun	+/- 1 %	Axial
Widerstandswert	Pull-up-Widerstand: 30 kΩ +/- 1 %, 0,25 W, mit flammhemmender Beschichtung		

Tabelle 9: Farbcodierung des Pull-up-Widerstandes, Quelle: Eigene Darstellung.

8.5 Definierte Regeln für das Leuchten der LEDs

Das LED-Board wird mit dem Arduino kompatiblem Board als *Shield* (aufsteckbare Zusatzplatine) verbunden, sodass über einen Stecker der digitale Luftfeuchtigkeits- und Temperatursensor DHT11 angeschlossen werden kann. Die Farben der LED's wurden zufällig ausgewählt.

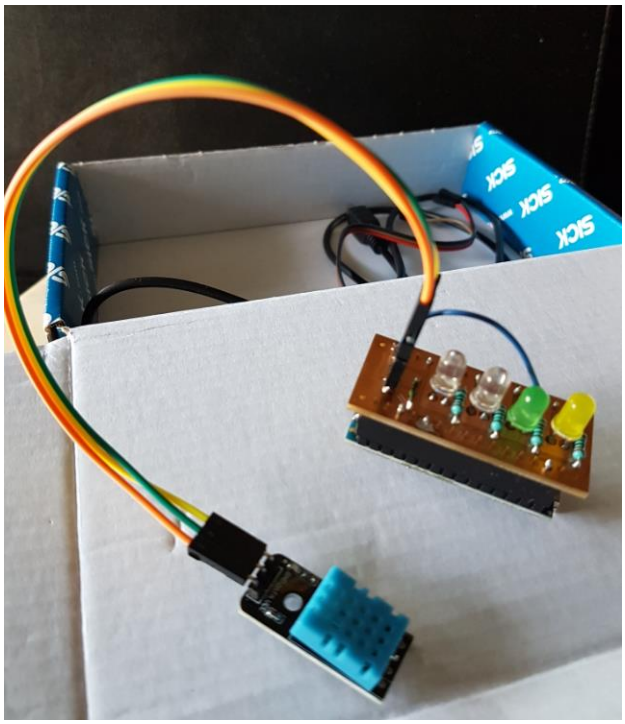


Abb. 38: Auf das Arduino kompatible Board aufgesetzte selbstdesignten Leiterplatte mit 4 LEDs und 220 Ohm Vorwiderständen, Quelle: Eigene Darstellung.

Die zwei klaren LEDs mit blauer Leuchtfarbe werden je nach Höhe des Messwertes der aktuellen Raumtemperatur mit einer Latenzzeit von ca. 2 sec und mit einer Messgenauigkeit von +/- 2 °C (diese Toleranzen sind bei Veränderungen der vom Sensor ermittelten Messwert und für weitere Auswertungen zu berücksichtigen) nach Folgenden im Programmcode definierten Regeln auf „*ON*" (= LED leuchtet) bzw. „*OFF*" (= LED leuchtet nicht) gesetzt.

Regeln für das Leuchten der blau-leuchtenden LEDs am LED-Board:

- Bei einer Raumtemperatur zwischen **0 – 9 °C** leuchtet keines der LED's.
- Bei einer Raumtemperatur zwischen **10 – 19 °C** leuchtet die erste blaue LED.
- Bei einer Raumtemperatur zwischen **20 – 29 °C** leuchtet die zweite blaue LED.
- Bei einer Raumtemperatur von **30 °C und darüber** leuchten beide blauen LEDs gleichzeitig.

Die blau-leuchtenden LEDs können nur über die Veränderung der Raumtemperatur (Messwerte die vom Sensor ausgelesen und mit einer Latenzzeit über den Mikrocontroller an den Rechner gesendet werden) beeinflusst werden. Eine manuelle Manipulation ist hier nicht angedacht und daher auch nicht vorgesehen. Ist das Arduino kompatible Board korrekt mit dem LED-Board und dem Sensor verbunden (zu achten ist auf die richtige Pinbelegung – siehe dazu auch Unterkapitel 6.2) werden die Daten ausgelesen und

gesendet. Demnach sollte bei Erreichen einer gewissen Minimaltemperatur (unter Berücksichtigung der Messgenauigkeit +/- 2 °C) zumindest immer eine blaue LED leuchten.

Die grün und gelb-leuchtenden LEDs können über die virtuelle Welt über einen in Unity 3D 2018 modellierten virtuellen Taster angesteuert und durch Betätigen des virtuellen Tasters mit beispielsweise der Computer-maus oder mit den Virtual Reality Controllern getrennt voneinander auf ON bzw. OFF gesetzt werden. Es ist im Programmcode definiert, dass jeweils eine der beiden LEDs auf ON gesetzt wird kann, oder beide gleichzeitig, oder keine von beiden leuchtet.

In Abb. 39 ist klar zu sehen, dass die zweite LED blau leuchtet (da die gemessene Raumtemperatur, wie in der vorigen Abbildung dargestellt, zum Zeitpunkt der Durchführung der Messung 23 °C beträgt). Zusätzlich wird über den virtuellen Taster über die Spielesoftware Unity 3D 2018 auch die gelb-leuchtende LED aktiviert und sie leuchtet daher ebenfalls.

Abb. 39: LED-Board mit einigen angesteuerten LEDs, Quelle: Eigene Darstellung.

8.6 Fehlerbehebung

Es werden nur 3 Anschlüsse (3 Pin-Stiftsockel) für den digitalen Sensor DHT11 (GND, DATA und VCC) benötigt, daher bleibt ein Steckplatz am LED-Board frei. Die Betriebsspannung beträgt 5 V (dies gilt es für die Bestückung der Platine mit den Halbleiterdioden zu beachten und ist der Grund, warum die blau-leuchtenden LED nur sehr matt, aber für das Testen der Anwendung dennoch ausreichend, leuchten.)

Bei der Erstellung des Platinenlayouts wird im Zuge der Umsetzung allerdings nicht berücksichtigt, dass der digitale Ein- und Ausgang D1 (TX1) der Datenübertragung zwischen dem Mikrocontroller und dem Computer dient. Daher muss dieser Pin am LED Board nachträglich entfernt werden um die Kommunikation nicht zu stören.

Am LED-Board wird daher nachträglich noch eine Korrektur vorgenommen, indem der Anschluss DATA mit einem Kabel (*Jumpwire*) auf den digitalen Ein- und Ausgang D10 verschaltet wird.

9 FUNKTIONEN, ANWENDUNG SOWIE EINRICHTUNG DER VIRTUAL REALITY-HARDWARE

In den folgenden Unterkapiteln werden die wesentlichen, für die Umsetzung der Ziele im Zuge der Erstellung dieser Masterarbeit, eingesetzten Hardwarekomponenten für das Arbeiten und Interagieren in und mit der virtuellen Welt analysiert. Die komplette Virtual-Reality-Hardware wurde im Vorfeld bereits von der FH CAMPUS 02 ausgewählt und wird für die Dauer der Programmierung und der Durchführung der Funktionstests zur Verfügung gestellt.

Das für die Erstellung dieser Masterarbeit verwendete VR-Hardware Set besteht neben einem DELL Alienware *Desktop-Gaming*-Rechner im Wesentlichen aus den folgenden Komponenten:

- 1 HTC Vive Virtual Reality-Brille
- 2 HTC Vive Virtual Reality-Controllern
- 2 HTC Basis-Stationen für Vive (Betriebssystem: *Steam*VR), welches zum Erfassen der räumlichen Position von Virtual Reality-Brillen und VR-Controllern sowie deren Synchronisation dient.
- 1 HTC Anschlussbox und
- 1 HTC Vive HDMI 3in1 Kabel (Länge: 5 m)

Abb. 40: Die HTC Vive VR-Hardware, Quelle: IT-Kurzzeitvermietung (2018), Online-Quelle [30.05.2019]

9.1 Die HTC Vive Virtual Reality-Brille

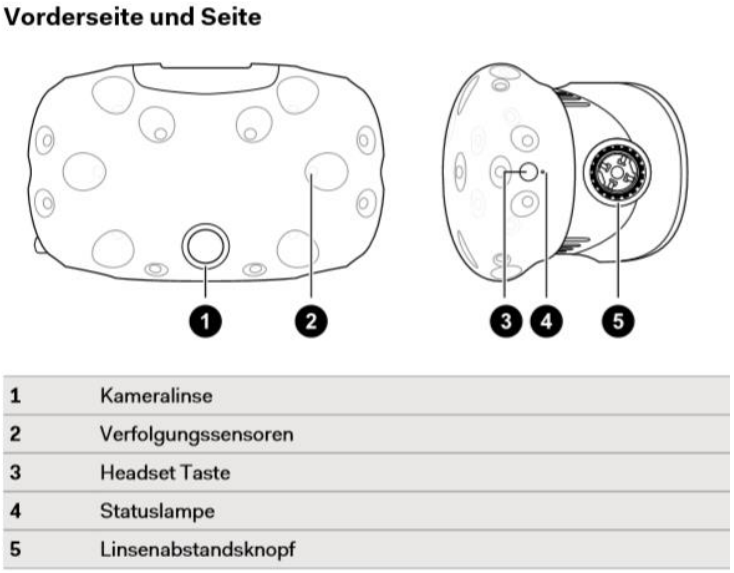
Abb. 41: Die HTC Vive VR-Brille, Quelle: Conrad Electronic (2018), Online-Quelle [30.05.2019]

HTC Vive ist aus einer Kooperation von HTC und Valve, der Firma, die hinter *Steam* steht, entstanden. HTC Vive ist zudem die erste *SteamVR*-Brille, die am Markt erschienen ist. Auf der Vive-*Website* stehen derzeit weit mehr als 2.000 Apps und Spiele *online* zur Verfügung.

Die HTC Vive Virtual Reality-Brille (siehe dazu auch Abb. 41), wird auch HTC Vive-*Headset* genannt. Sie hat bei 90 Hz eine Auflösung von 1.080 x 1.200 Pixel pro Auge, verfügt über zahlreiche Sensoren wie z.B. einen *Accelerometer* (Beschleunigungssensor), ein Magnetometer sowie einen Gyrosensor und hat ein Gewicht von ca. 561 g (ohne Kabel). Die Sensoren (siehe dazu auch Abb. 42) werden von den zwei Basis-Stationen *getrackt*.

Gemäß HTC Vive Handbuch ist eine Anwendung mit freier Bewegung im Raum auf einer Fläche von ungefähr 4,5 m x 4,5 m, sowie im Stehen auf einer Fläche von mindestens 2 m x 1,5 m (minimaler Bereich der zimmergroßen Einrichtung) als auch im Sitzen (keine Mindestflächenanforderung) möglich. Das *Chaperone*-System von Vive (siehe dazu auch Unterkapitel 11.3) warnt, wenn die Grenzen dieser Fläche erreicht werden. Das *SteamVR-Tracking* bietet eine sehr hohe Abdeckung der Bewegung auf der gesamten Fläche.

Der Linsenabstand an der Virtual Reality-Brille muss angepasst werden, wenn der Anwender eine Brille trägt. Dies sollte jedoch so gering wie möglich sein, denn je näher sich die Linsen am Auge befinden, desto besser ist das Sichtfeld beim Tragen der Virtual Reality-Brille.



1	Kameralinse
2	Verfolgungssensoren
3	Headset Taste
4	Statuslampe
5	Linsenabstandsknopf

Abb. 42: Vorderseite und Seitenansicht der HTC Vive Virtual Reality-Brille, Quelle: HTC Vive Handbuch (2018), Online-Quelle [04.06.2019]

Die HTC Vive Virtual Reality-Brille unterscheidet sich von der Oculus Rift dadurch, dass sie ein, auf Laser basierendes *Tracking-System* einsetzt. Die HTC Vive ist auch aufgrund des fehlenden *Thumbsticks* besser für *Room-Scale* als für *Standing* geeignet. Als *Standing* bezeichnet man die VR-Software, die man im Stehen verwendet und als *Room-Scale* die VR-Software, die darauf ausgelegt ist, dass man sich mit der Virtual Reality-Brille im Raum, auch beispielsweise dynamisch, bewegt.

9.2 Bedienung der HTC Vive Virtual Reality Motion-Controller (VR-Controller)

In der virtuellen Welt ist es neben den üblichen *Motion-Controllern*, wie z.B. einem Joystick oder einer Computermaus, auch möglich mit Hilfe von Virtual Reality-Controllern (siehe dazu auch Abb. 43) zu navigieren und sich in der Virtuellen Welt auf diese Weise fortzubewegen.

Unter *Motion-Controller* versteht man Eingabegeräte, die der Anwender für gewöhnlich in der Hand hält und deren Position und Rotation von der Software erkannt werden. Die Funktionstests im Zuge der Ausführung dieser Masterarbeit werden mit der HTC Vive Virtual Reality-Brille und den HTC Vive Virtual Reality-Controllern durchgeführt.

In den folgenden Abschnitten und Unterkapiteln wird ausschließlich auf einige wesentliche und für die Ausführung des praktischen Teiles dieser Masterarbeit relevante Buttons der Virtual Reality-Controller näher eingegangen und deren Funktion im Folgenden auch kurz beschrieben. Es ist wichtig sich mit der Virtual Reality-Hardware gut vertraut zu machen um die Softwareelemente in der virtuellen Welt gut und effizient ansteuern zu können.

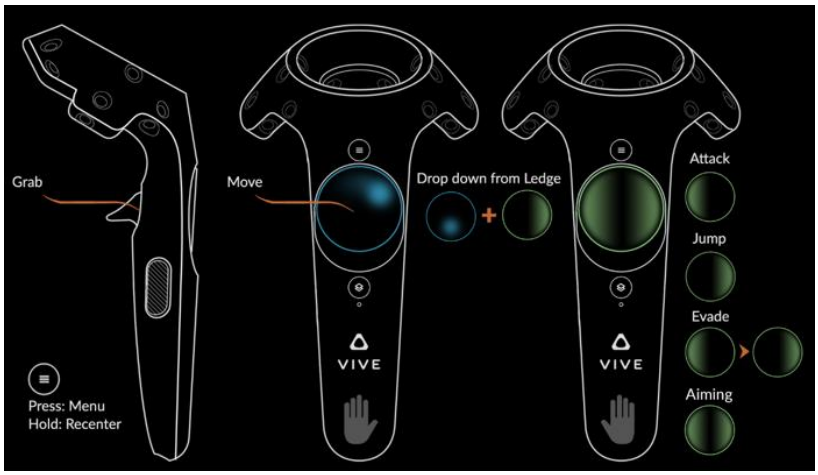


Abb. 43: HTC VIVE Controller, Quelle: STEAM Community (2018), Online-Quelle [30.04.2019].

Für die Bewegung in der virtuellen Welt wird das runde Touchpad (*Move*- bzw. *Trackpad*) auf der oberen Seite der zwei VR-Controller gedrückt (siehe dazu auch Abb. 43 und 44). Der *Move-Button* am Controller ist standardmäßig in vier Quadranten unterteilt. Dieser Button wird benötigt um die gewünschte Richtung in der Darstellung durch den Anwender selektieren zu können.

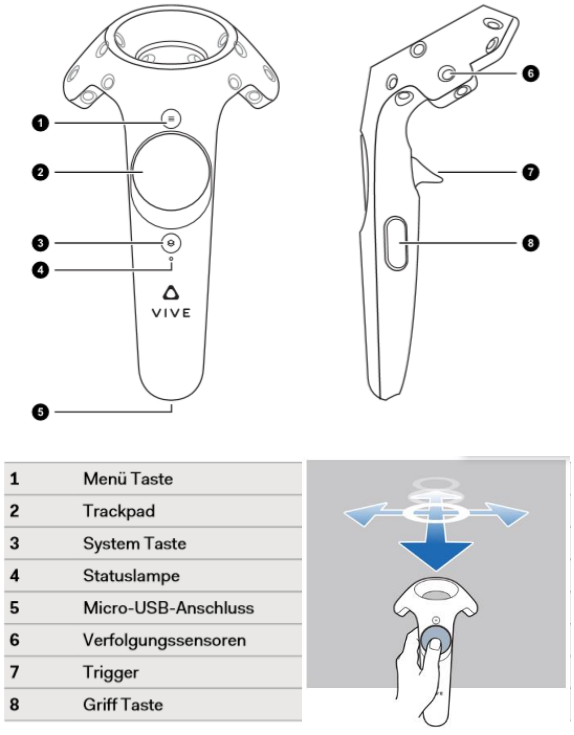


Abb. 44: Navigation der VR-Controller, Quelle: HTC Vive Handbuch (2018), Online-Quelle [04.06.2019]

Über ein optional verwendbares *Open-Source-Tool*, den „*Virtual-Button-Designer*“, können auch virtuelle Schaltflächen speziell für die Spielengine Unity 3D 2018 erstellt werden. Die virtuellen Schaltflächen werden in der virtuellen Welt für die leichtere Bedienung durch den Anwender auch visualisiert.[49]

[49] Vgl. Biagioli (2016), Online-Quelle [30.04.2019].

Durch die Bedienung des *Touchpads* kann der Anwender den gewünschten Punkt mit dem *Laserpointer* anvisieren. Ist die Farbe des Laserstrahles grün, dann ist es auch möglich eine Teleportation (*Locomotion*) durchzuführen. Wenn der Laserstrahl allerdings eine rote Farbe signalisiert, dann ist es nicht möglich eine Aktion mit diesem durch den Anwender auszuführen. Zum Hochheben oder Verschieben von Objekten, wird der *Trigger* auf der Rückseite der Controller gedrückt gehalten. Es gibt auch Objekte, die mit einmaligem Drücken des *Triggers* hochgehoben werden können. Nach dieser Systematik können in der virtuellen Welt beispielsweise auch Schubladen, Schiebetüren oder Eingangstüren geöffnet werden.

Um Objekte und z.B. Schalter zu bedienen, um das Licht über die virtuelle Welt auch in der realen Welt einzuschalten, Messwerte in der realen Welt auszulesen und in der virtuellen Welt darzustellen oder beispielsweise eine Webseite virtuell anzuwählen bzw. zu bedienen, müssen die Controller lediglich das Objekt in der virtuellen Welt berühren, um einen dahinter programmierten Code auszuführen.

9.3 Die Einrichtung der zwei HTC Basis-Stationen

Die zwei HTC Basis-Stationen ermöglichen das Eintauchen in die raumumfassende Virtuelle Realität. Mit ihnen wird die räumliche Position von Virtual Reality-Brille und Virtual Reality-Controllern erfasst. Dazu zählt auch die drahtlose Synchronisierung.

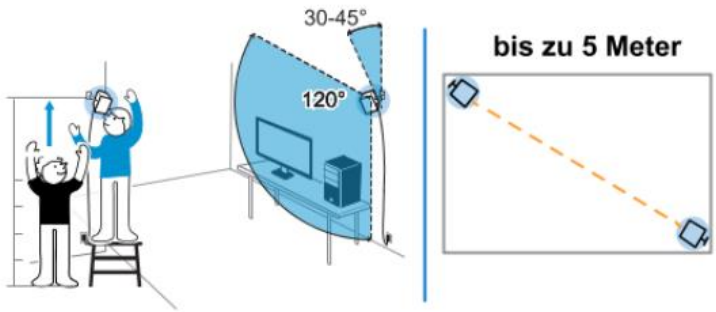


Abb. 45: Einrichtung der HTC Basis-Stationen, Quelle: Vive (2019), Online-Quelle [30.05.2019]

Auf der Website des Herstellers Vive sowie im HTC Vive Handbuch wird empfohlen die Basis-Stationen diagonal in entgegengesetzten Ecken, in einem Abstand von max. 5 m und in einer Höhe von ca. 2 m sowie mit einem Neigungswinkel von 30 bis 45 ° für eine genaue Bewegungsverfolgung nach unten ausgerichtet, anzubringen, sodass sich die beiden Stationen gegenseitig auch gut sehen (siehe dazu auch Abb. 45). Wenn man sich im „Spielbereich“ befindet, sollten die Virtual Reality-Brille und die Controller mindestens 0,5 m und nicht weiter als 5 m von den Basis-Stationen entfernt sein.

Dies wird sicherstellen, dass die Geräte richtig innerhalb des 120 ° Sichtfeldes der Basis-Station verfolgt werden. Somit soll sichergestellt werden, dass sie den gesamten Spielbereich gut abdecken. Es ist darauf zu achten, dass keine Hindernisse zwischen den beiden Basis-Stationen stehen. Sehen sich die Basis-Stationen nicht, dann müssen diese zusätzlich mit einem Synchronisationskabel verbunden werden. Die Basis-Stationen sollten in keinem Bereich mit hellem Licht aufgestellt werden, denn die Leistung der Basis-Stationen könnte dadurch negativ beeinflusst werden.

Sobald die HTC Basis-Stationen in Betrieb genommen werden sollte die Statuslampe grün leuchten und den Kanal anzeigen. Man sollte die Basis-Stationen nach dem Einschalten auch nicht mehr verschieben oder den Winkel ändern, da dies die Bewegungsverfolgung unterbricht. Anderenfalls muss man den Sichtbereich erneut einrichten.

10 SOFTWAREERSTELLUNG FÜR DIE INTERAKTION MIT DER VIRTUELLEN REALITÄT

10.1 Arduino IDE Entwicklungsumgebung, Version 1.8.4

Abb. 46: Arduino Logo, Quelle: Arduino (2019), Online-Quelle [30.04.2019].

Die Arduino-Plattform besteht, wie schon eingangs angeführt, aus einem Hardware- und einem Softwareteil. Die kostenlos verfügbare Entwicklungsumgebung, die auch Arduino IDE (*Integrated Development Environment*) (siehe auch Abb. 47). genannt wird, ist der Softwareteil. Die Programmiersprache ähnelt C/C++. Es wird über den USB-Port von der Entwicklungsumgebung in den Speicher des Arduino Boards geladen. Die Datenübertragung erfolgt seriell. Für die Erstellung von Programmen zur Kommunikation mit dem Arduino Nano kompatiblem Board wird die Entwicklungsumgebung „*Arduino Desktop IDE*“ am Computer installiert. Diese steht als *Open-Source-Software* auf der Arduino Homepage kostenlos zum *Download* zur Verfügung. Die physische Verbindung zum Hardwareteil muss während der Entwicklungs- und Testphase immer aufrechterhalten bleiben.

Die Arduino Software enthält einen Texteditor. Dieser dient zur Erstellung des Programmcodes. Darüber wird die Verbindung mit der Arduino Hardware hergestellt, um Programme zu übertragen und mit der Hardware zu kommunizieren. Ein Programm, das in *Arduino IDE* geschrieben wird, wird als *sketch* bezeichnet. Es steht in *Arduino IDE* auch eine Syntaxerkennung zur Verfügung Das Ausgabefenster (*message area*) dient als Informationsbereich für Statusmeldungen des *Compilers* und der Programmiersoftware. Er gibt Rückmeldung beim Speichern und Exportieren und zeigt Fehler, z.B. bei der Kompilierung von fehlerhaften Programmcodes an. Dabei springt neben der Anzeige der Fehlermeldung der *Cursor* in die Nähe der fehlerhaften Zeile und im Editor wird diese Zeile auch gelb markiert.

Mit Schaltflächen in der Symbolleiste können Programme überprüft und hochgeladen bzw. übertragen werden. Beim Übertragen des Programmes wird dieses kompiliert und auf den Mikrocontroller geladen. Über diese Schaltflächen kann auch der *Serial-Monitor* geöffnet werden um zu überprüfen, ob die Kommunikation richtig hergestellt wurde und Daten übertragen werden. Die erforderlichen Module wie Bibliotheken / Libraries werden ebenfalls in der Entwicklungsumgebung verwaltet.

Ein *sketch* teilt dem Mikrocontroller mit, welche Aufgaben durchzuführen sind. Ein Programmierprojekt wird in einzelne Programmschritte (Befehle bzw. Kommandos) unterteilt, die in einer bestimmten Reihenfolge abgearbeitet werden. Der Flash-Speicher übernimmt die Aufgabe der Ablage. Wird der Arduino von der Spannungsversorgung getrennt, dann bleibt der *sketch*, der auf den Mikrocontroller übertragen wurde, resistent im Speicher. Das EEPROM ist ebenfalls ein nichtflüchtiger Speicher, der einmal gespeicherte Daten auch nach dem Verlust der Versorgungsspannung behält. Er kann dazu genutzt werden, z.B. Messwerte permanent zu speichern.[50]

[50] Vgl. Bartmann (2017), S. 29.

Die Entwicklungsumgebung muss von der Arduino-Webseite heruntergeladen und installiert werden.

In der nachfolgenden Abb. 47 wird als Statusinformation und im Ausgabefenster eine Fehlermeldung ausgegeben. Diese Fehlermeldung bedeutet, dass das Arduino kompatible Board nicht angeschlossen ist und daher eine Kompilierung nicht möglich ist.

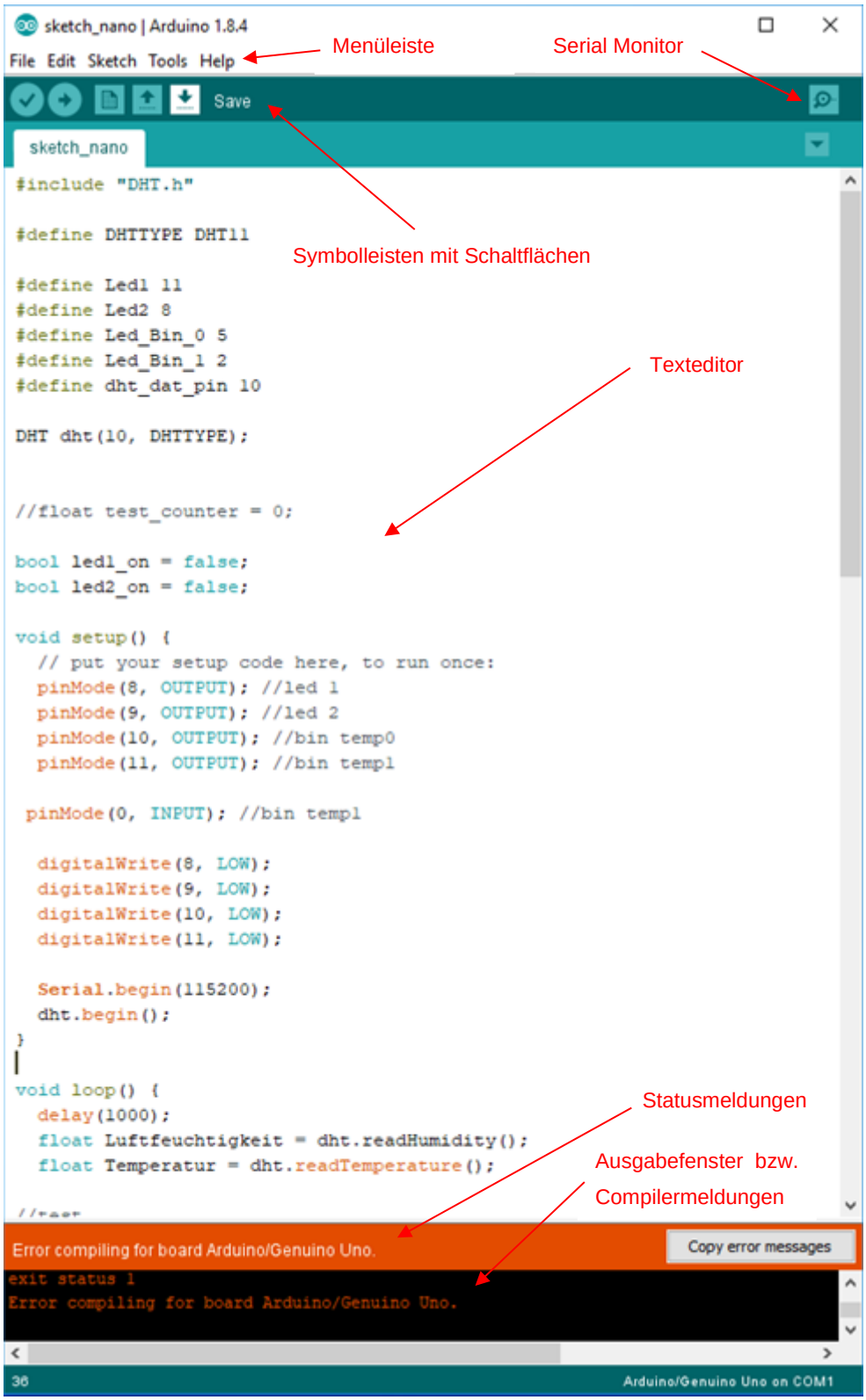


Abb. 47: Arduino Entwicklungsumgebung IDE, Quelle: Eigene Darstellung.

In der Symbolleiste befinden sich Schaltflächen wie „Kompilieren“ (Übersetzen in eine für den Mikrocontroller verständliche Maschinensprache) des Quellcodes, das Hochladen des übersetzten Codes zum Mikrocontroller oder das Laden und Speichern von *sketches*.

Wenn der Serielle Monitor (*Serial Monitor*) geöffnet wird, kann auf die serielle Schnittstelle zugegriffen werden, um dort auf diese Weise Informationen anzuzeigen oder auch zu versenden.

Der *Compiler* hat die Aufgabe, den Quellcode zu übersetzen. Meldungen über den Fortschritt oder aufgetretene Probleme und abschließend auch Hinweise über den Speicherverbrauch werden in der Entwicklungsumgebung als Compiler-Meldung ausgegeben.

Die Statusinformation kann grundsätzlich z.B. ein Hinweis darüber sein, was gerade passiert oder Meldungen der zuletzt ausgeführten Aktionen.[51]

10.2 Arduino-Softwareprogramm (sketch_nano)

Das Arduino-Softwareprogramm (Dateiname: *sketch_nano*) besteht im Wesentlichen aus Kommentaren, Compiler-Anweisungen, Deklarationen, Definitionen, Ausdrücken, Zuweisungen und Funktionen. Bei einem Arduino-Softwareprogramm (*sketch*) fehlt grundsätzlich die z.B. bei C obligatorische *main*-Funktion.

Bei allen C-Compilern wird bereits eine Menge von nützlichen Funktionen in Form von Bibliotheken mitgeliefert (z.B. für die Ein- bzw. Ausgabe, *String*-Verarbeitung, usw.). Man „versteckt“ diese in sogenannten Bibliotheken (*Libraries*). Damit der *Compiler* diese beim Übersetzen des C-Quelltextes auch findet, muss man diesem mitteilen, dass er die Dateien, genannt *Header*-Dateien, miteinschließen soll (*#include*). *Header*-Dateien erkennt man an der Endung .h.[52]

Für die Programmierung ist die Pinbelegung (digitale Ein/Ausgänge D11 und D08) relevant. Für die z.B. gelb-leuchtende LED1 (Vorwiderstand R5) und die grün-leuchtende LED2 (Vorwiderstand R4) ist die folgende Abbildung relevant und gilt analog für die zwei blau-leuchtenden LEDs, Bin_0 auf D05 (Vorwiderstand R3) und Bin_1 auf D02 (Vorwiderstand R2).

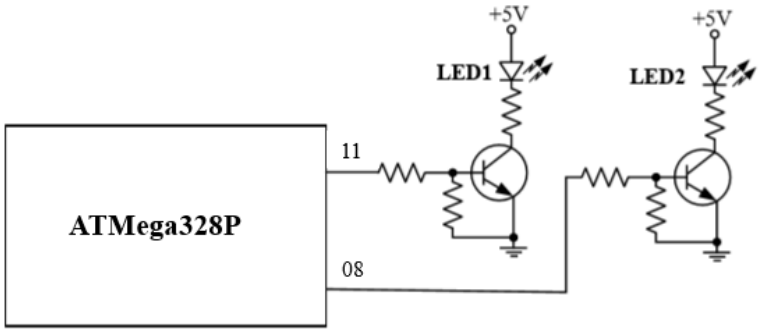


Abb. 48: Pinbelegung des LED- und Arduino kompatiblen-Boards, Quelle: Eigene Darstellung.

[51] Vgl. Bartmann (2017), S. 52.

[52] Vgl. Schalk (2019), S. 23.

Vor der Erstellung des Softwareprogrammes sind die folgenden wesentlichen Hintergrundinformationen zur Programmierung in C/C++ erforderlich. Die Sprache C unterscheidet sich dabei nur geringfügig von C++.

10.2.1 Variablendeklaration

Vor der ersten Verwendung einer Variablen muss diese deklariert, definiert und initialisiert werden: [53]

- Deklaration: Bei der Deklaration teilt man dem Compiler den Namen und den Datentyp einer Variablen mit. Es wird jedoch kein Speicher reserviert. Mit dem Schlüsselwort *extern* wird eine Variable z.B. nur deklariert.
- Definition: Erst bei der Definition wird der Speicher reserviert.
- Initialisierung: Bei der Initialisierung wird der Variablen ein bestimmter Wert zugewiesen. Vor dem ersten Lesezugriff muss man die Variable immer initialisieren, ansonsten liest man einen nicht definierten Wert aus dem Speicher. Wird ein *sketch* zum Beispiels zur Verarbeitung von Messwerten benötigt, dann müssen diese Werte in irgendeiner Form innerhalb des Mikrocontrollers abgelegt werden. Dazu nutzt man diese Variablen, die dann als Platzhalter agieren. Es handelt sich um spezielle Speicherbereiche für den Datenaustausch und zur Datenmanipulation. Diese Speicherbereiche sind jedoch flüchtig, was bedeutet, dass sie ihre Informationen nach dem Abschalten der Spannungsversorgung verlieren. Ein *sketch* nutzt diese Daten nur zur Laufzeit und sie werden nach dem erneuten Starten wiederhergestellt. [54]

In den meisten Beispielen erfolgt sowohl die Deklaration und Definition in einem Schritt. Um einer Variablen einen Wert zuzuweisen, wird der Zuweisungsoperator „=“ verwendet. Einer Variablen kann nur ein Wert eines Datentyps zugewiesen werden, den sie selbst besitzt.

10.2.2 Konstanten und Präprozessor-Konstanten

Konstanten bezeichnen Variablen, welche im Programm nicht verändert werden können. Ein typisches Beispiel ist die Definition für einen Port-Pin. Konstanten werden verwendet um die Lesbarkeit eines Programmes zu erhöhen. Schließt man z.B. eine LED an den Port-Pin 11, so ist es erforderlich, in allen Programmteilen diese Nummer zu verwenden. Allerdings kann der Wert 11 auch eine ganz andere Bedeutung haben. Daher soll man einfach eine Konstante wie z.B. „*led1*“ verwenden.[55]

Besonders interessant ist die Anwendung von Präprozessor-Kontanten. Mit der Präprozessor-Direktive *#define* werden grundsätzlich Makros und Konstanten definiert. Vor dem eigentlichen Kompilieren ersetzt der Präprozessor im C-Quellcode (z.B. die Konstanten „*led1*“ gegen den Wert 11). Der Vorteil ist, dass kein zusätzlicher Speicherplatz im Mikrocontroller für die Konstante benötigt wird. Präprozessor-Konstanten besitzen jedoch keinen Typ. Dadurch kann der Compiler keine Typ-Fehler erkennen.[56]

---

[53] Vgl. Schalk (2019), S. 27.

[54] Vgl. Bartmann (2017), S. 29.

[55] Vgl. Schalk (2019), S. 30.

[56] Vgl. Schalk (2019), S. 30 - 31.

10.2.3 Void setup () – void loop ()

```
void setup()
{
  anweisungen;
}

void loop()
{
  anweisungen;
}
```

Abb. 49: Struktur der C-Sprache in Arduino IDE
Quelle: Arduino Programmier-handbuch (2018), Online-Quelle [03.06.2019].

Im Arduino-Programmierhandbuch ist angeführt, dass der grundlegende Aufbau der Arduino Programmiersprache in C/C++ sich in mindestens zwei Teile aufteilt. Diese zwei Teile oder Funktionen umschließen Blöcke von Anweisungen.

Hierbei dient *setup()* als Funktions-Vorbereitung und *loop()* dient der eigentlichen Funktions-Ausführung. Beide Funktionen sind notwendig damit das Programm auch ausgeführt werden kann.

Die Variablen-Definition sollte immer vor der *Setup*-Funktion stehen. *Setup* muss immer als erste Funktion in einem Programm durchlaufen werden. Es wird nur einmal ausgeführt und dient z.B. dem Setzen von *PinMode* oder der Initiierung der seriellen Kommunikation.

Nach der *setup()* Funktion folgt die *loop()* Funktion. Sie beinhaltet den Programmcode, der kontinuierlich in einer unendlichen Schleife ausgeführt wird (z.B. Auslesen der Eingänge, Triggern der Ausgänge, etc.). Diese Funktion stellt den Kern aller Arduino-Programme dar.

10.2.4 Schlüsselwörter

Im ANSI-Standard der Sprache C sind insgesamt 32 sogenannte „Schlüsselwörter“ definiert (siehe dazu auch Abb. 50). Für C++ sind 11 zusätzliche Schlüsselwörter zu beachten.

Keywords in C Programming			
auto	break	case	char
const	continue	default	do
double	else	enum	extern
float	for	goto	if
int	long	register	return
short	signed	sizeof	static
struct	switch	typedef	union
unsigned	void	volatile	while

Abb. 50: 32 Schlüsselwörter der Sprache C/C++, Quelle: LEARN C (2019), Online-Quelle [03.06.2019].

Diese 32 Schlüsselwörter sind für den Compiler reserviert. Alle Schlüsselwörter müssen in Kleinbuchstaben geschrieben werden und dürfen nicht für andere Zwecke wie z.B. für Variablennamen, Funktionsnamen, Klassen usw. verwendet werden. Darüber hinaus sind Operatornamen (z.B. not !, and &&, xor ^, not_eq !=, etc.) für einige Operationen wie Schlüsselwörter zu behandeln.

Der Datentyp *void* wird in der Sprache C/C++ als „unvollständiger Typ“ bezeichnet. Er darf nicht als Variablenname verwendet werden. Diesen Datentypen benötigt man dazu, wenn eine Funktion keinen Wert

zurückgeben soll, für die Deklarationen einer leeren Parameterliste für eine Funktion oder als Teil des regulären aber anonymen Datenzeigertyps *void**, der Zeiger aller Datentypen (keine Funktionen) aufnehmen kann.

10.2.5 Switch-Anweisung

Mit der switch-Anweisung kann eine Mehrfachverzweigung realisiert werden. Der Wert des switch-Ausdrucks wird mit den Konstanten bei den case-Marken verglichen, anschließend wird zur entsprechenden Marke verzweigt und die entsprechende Anweisung ausgeführt. Der ganzzahlige Ausdruck nach dem Schlüsselwort *switch* entscheidet also, welcher Fall (*case*) einer Mehrfachverzweigung abgearbeitet wird.

Wenn keine passende Marke existiert, wird bei der Marke *default* weitergearbeitet, falls die vorhanden ist, sonst nach der switch-Anweisung.[57]

10.2.6 Eingangs-/Ausgangs-Ports

Eingangs-/Ausgangs-Ports (Input/Output-Ports bzw. abgekürzt *I/O-Ports*) moderner Mikrocontroller können wahlweise als Eingang- oder als Ausgang verwendet werden. Deshalb werden diese auch als GPIOs (*General Purpose Input Output*) bezeichnet. Ob ein Port-Pin als Eingang oder als Ausgang verwendet wird, legt man bei der Programmierung der entsprechenden Konfigurations-Register (*Special Function Register* bzw. abgekürzt SFR) fest.[58]

10.2.7 Schrittgeschwindigkeit (Baudrate) und Übertragungsgeschwindigkeit

Die Schrittgeschwindigkeit gibt an, wie viele Zustandsänderungen pro Sekunde (Einheit in Baud) auf dem Übertragungskanal stattfinden.

Die Übertragungsgeschwindigkeit gibt an, wie viele Bits pro Sekunde (BPS) übertragen werden. Bei Signalen, die pro Schritt immer nur 1 Bit übertragen, ist die Schrittgeschwindigkeit identisch mit der Übertragungsgeschwindigkeit.

10.2.8 Arduino-Library-Manager

Die DHT *Sensor Library* ist nicht Bestandteil der in der Arduino IDE integrierten Software-Library. Deshalb ist es erforderlich, diese Library nachträglich zu installieren (siehe dazu auch die nachfolgenden Abb. 51 und Abb. 52).

Der Sensor DHT11 besitzt lediglich drei Anschlüsse und wird über eine kleine Trägerplatine an den Arduino angeschlossen, auf dem sich schon der erforderliche Widerstand befindet. Er wird über das *One-Wire-Protokoll* angesteuert.

[57] Vgl. Schalk (2019), S. 40.

[58] Vgl. Schalk (2019), S. 70.

Aufrufen des Arduino Library Managers zur Installation der *DHT sensor library for DHT11*:

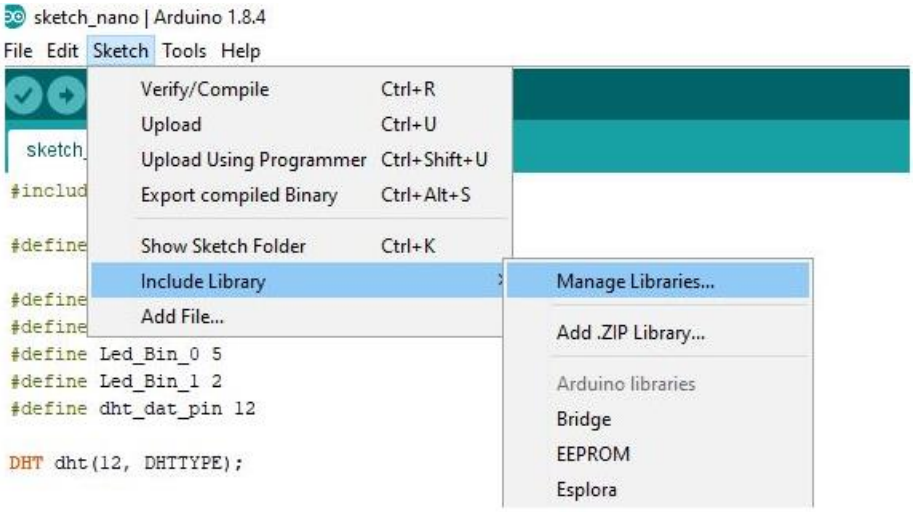


Abb. 51: Starten des Arduino Library Managers, Quelle: Eigene Darstellung.

Nachdem der Temperatur- und Luftfeuchtigkeitssensor DHT11 am Arduino Board angeschlossen wurde muss der Anwender im Suchfeld der Entwicklungsumgebung nur mehr „DHT“ eingeben und die markierte Library auswählen sowie den „*Install*“-Button betätigen.

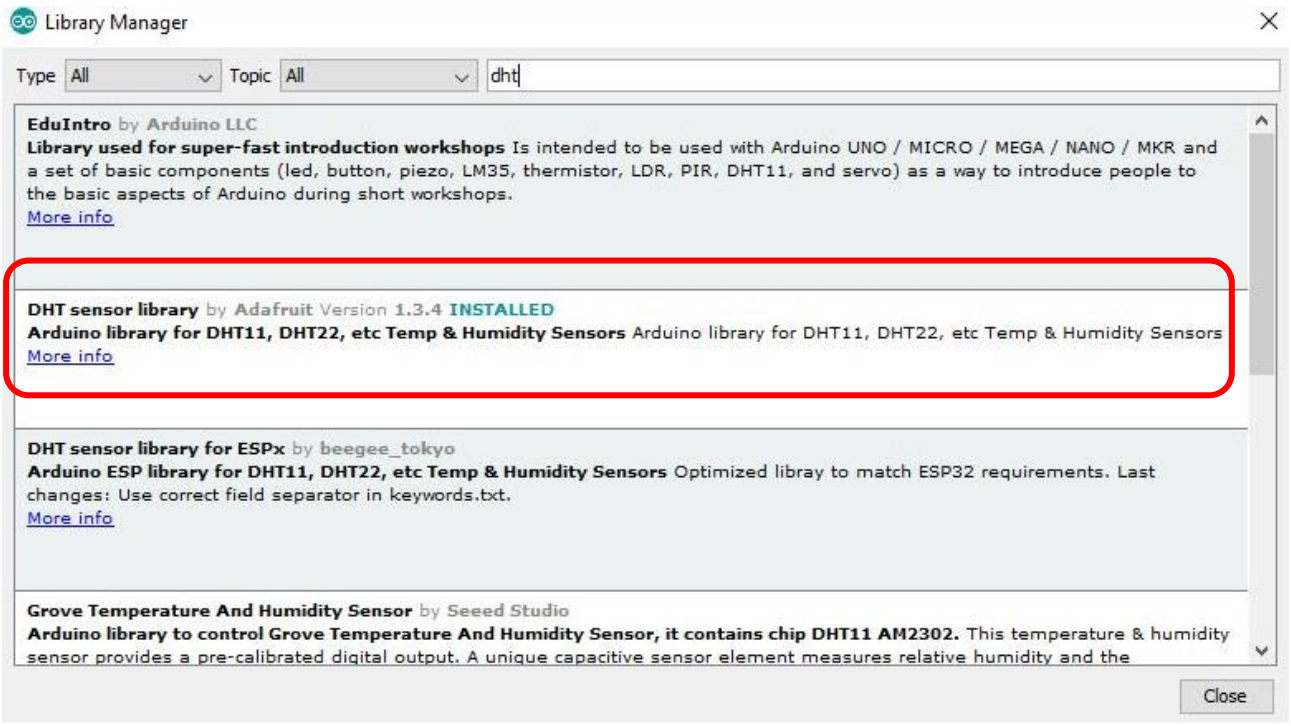


Abb. 52: Arduino Library Manager, Quelle: Eigene Darstellung.

Anschließend wird die Library automatisch aus dem Internet geladen und in der Entwicklungsumgebung installiert.

10.2.9 Umsetzung des Programmcodes (sketch)

In der folgenden Tabelle ist das Arduino-Softwareprogramm (*sketch*), mit einigen Kommentaren sowie direkten Anmerkungen dargestellt. Weitere relevante Informationen können den vorigen Unterkapiteln entnommen werden.

Softwareprogramm	Anmerkungen
`#define DHT` `#define DHT11 11` `#define OUTPUT 1` `#define INPUT 0` `#define LOW 0` `#define HIGH 1` `int dht(int t1, int t2)` `{` `}` `int pinMode(int t1, int t2)` `{` `}` `int digitalWrite(int t1, int t2)` `{` `}` `void delay(int t)` `{` `}` `bool isnan(int t)` `{` `}` `void setBinaryTemp(float temp);` `void readPcCommand();` `class Serial` `{` `  void begin(char* str);` `}`	
`#include "DHT.h"`	Präprozessor-Anweisung
`#define DHTTYPE DHT11`	Bezeichnung des Luftfeuchtigkeitssensors DHT11
`#define Led1 11 //Pinbelegung: led1 auf Port-Pin 11` `#define Led2 8 //Pinbelegung: led2 auf Port-Pin 08` `#define Led_Bin_0 5 //Pinbelegung: led_bin_0 auf Port-Pin 05` `#define Led_Bin_1 2 //Pinbelegung: led_bin_0 auf Port-Pin 02`	Definition der LEDs
`#define dht_dat_pin 10` `DHT dht(10, DHTTYPE);`	
`//float test_counter = 0;`	// Einzeilige Kommentare
`bool led1_on = false;` `//Deklaration, Definition und Initialisierung`	

```
bool led2_on = false;


void setup()
{
        //put your setup code here, to run once:

        pinMode(8, OUTPUT); //led 1 - Ausgang setzen
        pinMode(9, OUTPUT); //led 2 - Ausgang setzen
        pinMode(10, OUTPUT); //bin temp0 - Ausgang
setzen
        pinMode(11, OUTPUT); //bin temp1 - Ausgang
setzen

        pinMode(0, INPUT); //bin temp1

        digitalWrite(8, LOW);
        digitalWrite(9, LOW);
        digitalWrite(10, LOW);
        digitalWrite(11, LOW);


        Serial.begin(115200);
        dht.begin();

}

// Die Funktion loop wird permanent in einer
Endlosschleife wiederholt.
void loop()
{
        delay(1000);
        //1.000 ms warten
        float Luftfeuchtigkeit = dht.readHumidity();
        //Nachkommastelle
        float Temperatur = dht.readTemperature();
        //Gleitkommazahl (Nachkommastellen)

        //test
        //test_counter = ++test_counter > 100.0 ? 0 :
test_counter;
        //Temperatur = Luftfeuchtigkeit =
test_counter;
        //test ende

        if (!isnan(Temperatur))
        {
            Serial.print('T');
            Serial.println(Temperatur);
        }

        if (!isnan(Luftfeuchtigkeit))
        {
            Serial.print('H');
            Serial.println(Luftfeuchtigkeit);
        }

        setBinaryTemp(Temperatur);

        readPcCommand();
}
```

Einfache Kommentare werden mit einem // am Anfang der Zeile definiert und enden mit dem Ende der Zeile. Sie werden vom Programm ignoriert und verbrauchen keinen Speicherplatz.

AUS: setzt den jeweiligen Ausgang auf LOW

begin()
Legt die Datenrate in Bit pro Sekunde (Baud) für die serielle Datenübertragung fest. Für die Kommunikation mit dem Computer sollte man eine der folgenden Raten: 300, 600, 1.200, 2.400, 4.800, 9.600, 14.400, 19.200, 28.800, 38.400, 57.600 oder 115.200 wählen.

Das Makro isnan() liefert einen Wert ungleich Null (d.h. *true*), wenn sein Argument eine NaN oder "not a number" ist.

`` void readPcCommand() { if (Serial.available() > 0) { //read the incoming byte: int incomingByte = Serial.read(); switch(incomingByte) { // switch led 1/2 on/off case '1': led1_on = !led1_on ;digitalWrite(Led1, led1_on); break; case '2': led2_on = !led2_on ; digitalWrite(Led2, led2_on); break; default: break; } } } void setBinaryTemp(float temp) { if (temp < 0) return; if ((temp >= 0) && (temp <10)) { digitalWrite(Led_Bin_0, LOW); digitalWrite(Led_Bin_1, LOW); } if ((temp >= 10) && (temp <20)) { digitalWrite(Led_Bin_0, LOW); digitalWrite(Led_Bin_1, HIGH); } if ((temp >= 20) && (temp <30)) { digitalWrite(Led_Bin_0, HIGH); digitalWrite(Led_Bin_1, LOW); } if (temp >= 30) { digitalWrite(Led_Bin_0, HIGH); digitalWrite(Led_Bin_1, HIGH); } } ```	Switch-Anweisung für Mehrfachverzweigung if-Bedingung Temperatur: Definition der Regeln für das Leuchten der blau-leuchtenden LEDs am LED-Board.

Tabelle 10: Softwareprogramm der Arduino IDE, Quelle: Eigene Darstellung.

11 SPIELEENGINE UNITY 3D 2018

Die Unreal Engine 4 verwendet C++ für die Programmierung und Unity 3D 2018 greift auf *JavaScript* oder C# zurück. Die Entscheidung welches Programm besser geeignet ist, wurde anhand der eigenen *Skill-Sets* beurteilt und ausgewählt.

Beide *Engines* verfügen auch über eigene *Asset Stores*. Unity 3D 2018 schneidet jedoch aufgrund der größeren Anzahl der *Assets* im *Asset Store Shop* wesentlich besser ab als die Unreal Engine. Zudem ist Unity 3D 2018 bei den Anwendern für seine einfach zu bedienende Benutzeroberfläche bekannt.

11.1 Umgang mit und Steuerung von Unity 3D 2018

Neue Projekte können beim Starten von Unity 3D 2018 über das Auswahlfeld *new* in der Projekt-Auswahl gestartet und mit *open* können bestehende Projekte von der Festplatte oder über die Cloud wieder geöffnet werden. Unity 3D 2018-Services kann an dieser Stelle ebenfalls ein- bzw. ausgeschaltet werden.

11.1.1 Editor in Unity 3D 2018

Der Unity 3D-*Editor* kann in fünf Bereiche im *User-Interface* untereilt werden. Das *Layout* des *Editors* kann durch *Drag & Drop* vom Anwender verändert werden. Anhand der nachfolgenden zwei Abbildungen werden die einzelnen Bereiche kurz vorgestellt.[59]

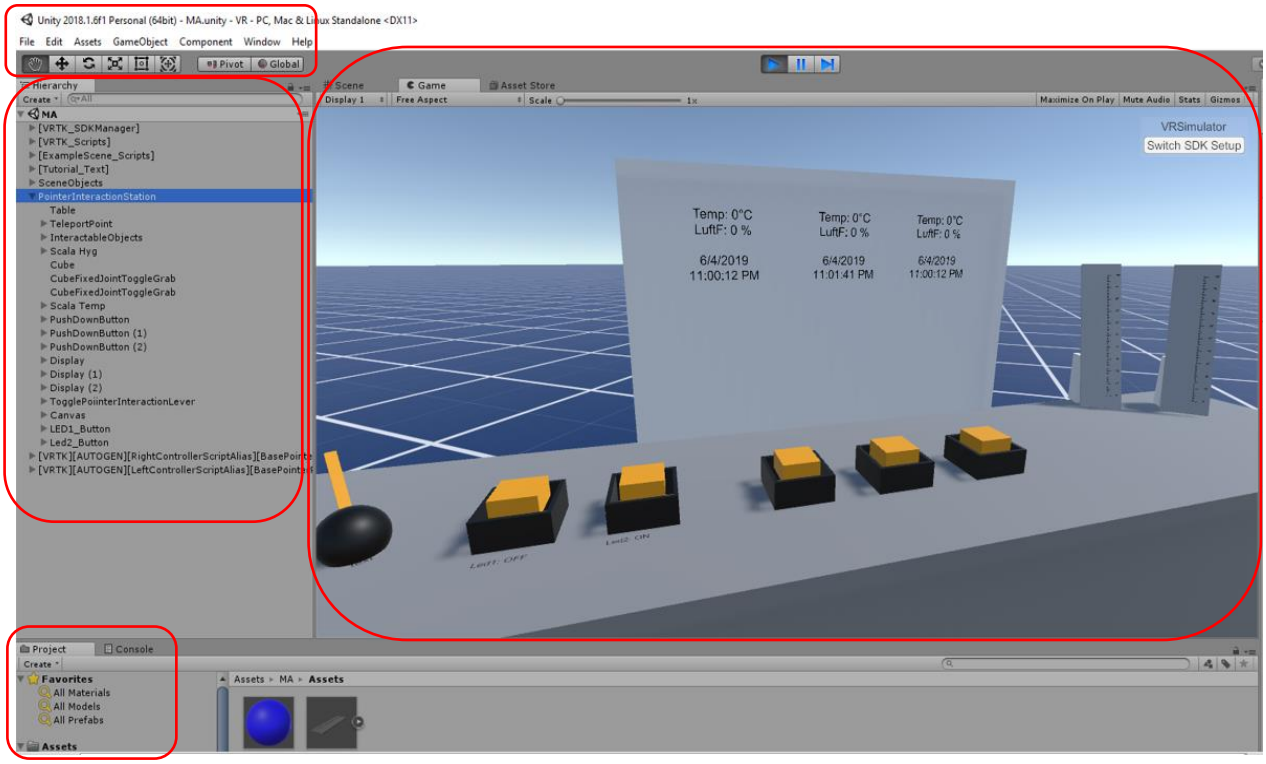


Abb. 53: Ansicht des Unity 3D 2018-Editors, Quelle: Eigene Darstellung.

[59] Vgl. Korgel (2018), S. 40 ff.

Project- und Console-Reiter:

Standardmäßig ist der Reiter *Project* aktiviert. Er stellt die Projektansicht (mit allen notwendigen Farben und Formen) dar, in der alle Dateien (*Assets*), die zu einem Projekt gehören angezeigt werden. Dieser Bereich wird auch *Project Browser* genannt. Über Rechtsklick in diesen Bereich und durch das Öffnen der Kontextmenüs *Create* sowie Folder können einzelnen Ordner (*folder*) neu angelegt und Szenen abgespeichert werden.

Im *Console*-Reiter werden während der Testphase einer Anwendungsentwicklung mögliche *Debug*-Informationen, Warnungen (*Warnings*) und Fehler (*Bugs, Errors*) oder *Exceptions* angezeigt. *Scripts* können hier jederzeit sogenannte *Debug-Logs* hineinschreiben.

Hierarchy:

Unter *Hierarchy* werden alle *Game Objects* der aktuellen Szene angezeigt. Jedes Modell, jedes Licht (z.B. das *directional light*) und jede Kamera (z.B. die *main camera*) erhalten hier einen ent-sprechenden Eintrag und können dadurch auch wiedergefunden und ausgewählt werden.

Über *Drag & Drop* kann der Anwender *Game Objects* umsortieren und hierarchisch anzeigen lassen. Eine gute Sortierung ist für den Überblick wichtig, vor allem bei sehr vielen sehr kleinen Objekten. Zieht man ein 3D-Modell aus dem *Project Browser* in die *Scene View*, wird automatisch ein entsprechendes *Game Object* angelegt, welches bereits alle *Components* besitzt, damit das 3D-Modell in der *Scene* richtig angezeigt wird. Das dadurch angelegte *Game Object* sieht der Anwender dann automatisch in der *Hierarchy*. Die in der *Hierarchy* markierten Objekte werden in der *Scene View* dann farblich hervorgehoben markiert und im *Inspector* (siehe dazu auch Abb. 55) findet der Anwender die Details zum markierten Objekt dazu übersichtlich angezeigt.

Unter *Create* können eine Vielzahl von vorkonfigurierten *Game Objects* erstellt werden. Dazu gehören zum Beispiel geometrische Formen wie Würfel oder Kugeln, aber auch Lichter, Audio-Quellen und Kameras.

Parenting, oder auch „Eltern-Kind-Beziehung“, ist in Unity 3D 2018 ein wichtiges Konzept zum Erstellen von Szenen. Man kann jedem *Game Object* beliebig viele andere *Game Objects* als Kinder zuweisen. Alle Kinder sind abhängig von ihrem *Parent*. Das bedeutet, wenn sich der *Parent* bewegt, dreht oder seine Größe ändert, ändern sich auch die Kinder entsprechend. Die Kinder können sich aber zusätzlich noch, immer relativ zu dem *Parent*, selber bewegen, drehen oder skalieren.

Scene-, Game- und *Asset-Store*-Reiter:

Dieser Bereich hat standardmäßig drei Reiter. Der *Scene*-Reiter ist dabei der wichtigste von diesen dreien. Hier erhält man in Echtzeit eine Vorschau der aktuellen Szene (*Scene*). Diese Ansicht wird daher auch als *Scene View* bezeichnet. Unter dem *Scene*-Bereich versteht man in Unity 3D 2018 also konkret die Umgebung, in der man die Anwendung projiziert. Diese ist standardmäßig dreidimensional. Alternativ kann der Anwender diese auf zweidimensional (also Darstellung in einer Ebene) schalten.

Die *Game*-Ansicht zeigt während der Testphase einer Anwendung die Perspektive, wie sie auch der Anwender in der fertigen Anwendung hat. Diese Ansicht wird auch *Game View* genannt. Der *Asset-Store* ist ein integrierter *Store*, in dem man Modelle, *Sounds* und *Plug-ins downoaden* kann. Teilweise stehen diese allerdings nicht kostenlos zur Verfügung, sondern müssen käuflich erworben werden.

Toolbar- bzw. Werkzeug-Bereich:

Dieser Bereich enthält die klassische *Toolbar* (*File*, *Edit*, *Assets*, *Game Objects*, usw.) sowie die Unity-*Toolbar*.

Die nachfolgende Abbildung zeigt den virtuellen Messtisch (der als Dummy für einen *Digital Twin* dient) in einem virtuellen Raum und im Hintergrund einen „Himmel“. Diese *Skybox* wird eingeschaltet, um die gesamte Szene zu rendern und um den Eindruck einer komplexen Szenerie am Horizont zu erwecken.

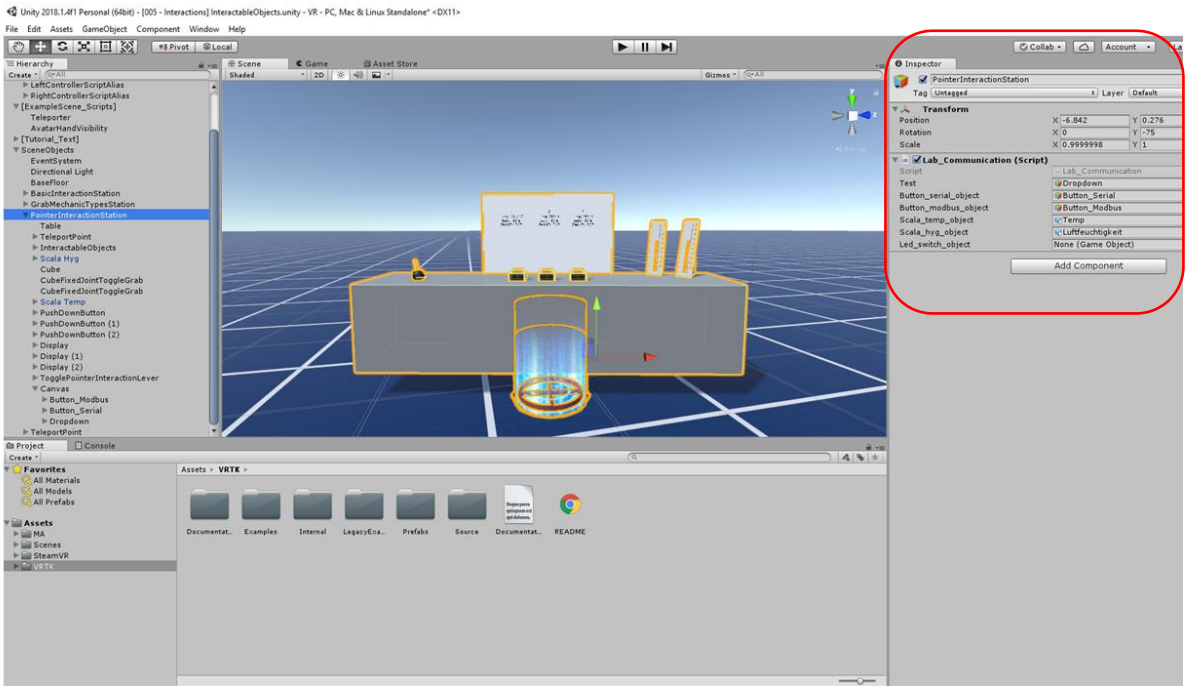


Abb. 54: Ansicht des Unity 3D 2018-Editors, Quelle: Eigene Darstellung.

Der *Inspector*-Bereich (siehe dazu auch Abb. 55) ist ein dynamischer Bereich, dessen Inhalt sich immer dem anpasst, was man gerade in Unity 3D 2018 ausführt. Hat der Anwender in der *Scene View* oder *Hierarchy* ein *Game Object* ausgewählt, zeigt er beispielsweise detaillierte Informationen bzw. alle möglichen Eigenschaften zu dem *Game Object* an.

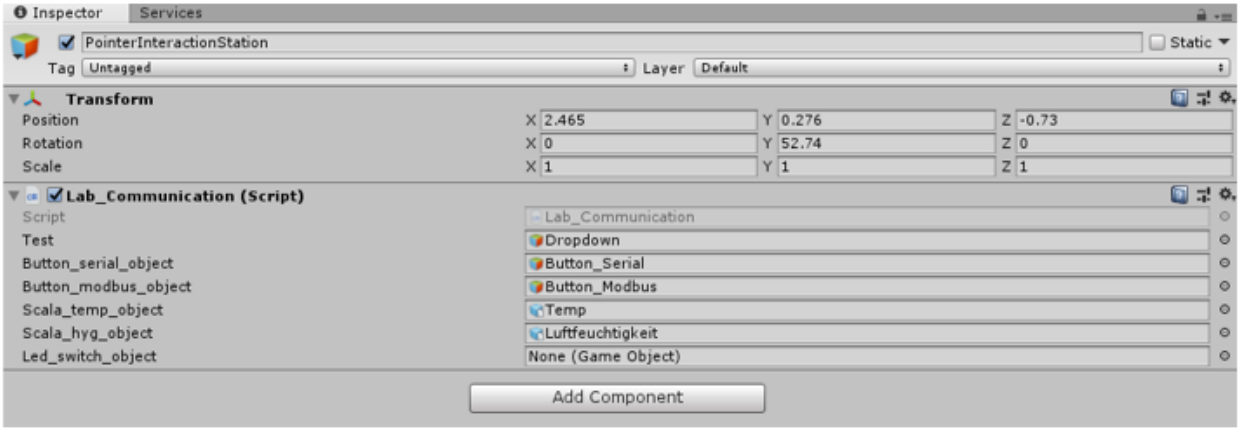


Abb. 55: *Inspector*-Bereich in Unity 3D 2018-Editors, Quelle: Eigene Darstellung.

In diesem Bereich werden jedoch auch diverse Einstellungsfenster angezeigt, wenn der Anwender sie über die Toolbar abruft. Dieses kontextsensitive Fenster ist auch der Ort, an dem man so gut wie jede Einstellung vornehmen kann. Die *Camera* zeigt zudem einen *preview* von dem, was durch die Virtual Reality-Brille oder am Computer-Monitor dann auch zu sehen ist.

11.1.2 Steuerung in Unity 3D 2018

In Unity 3D 2018 gibt es mehrere Formen von Eingabe-*Devices* und somit verschiedene Möglichkeiten zur Steuerung. Einerseits kann man während der Entwicklungsphase mit einer herkömmlichen Computer-Tastatur und Computer-Maus arbeiten und andererseits können die Tests, neben den vorhin genannten Geräten, auch mit einer Virtual Reality-Brille und Virtual Reality-Controllern durchgeführt werden. In der Entwicklungsphase kann der Anwender in Unity 3D 2018 mit der Computer-Tastatur (*Keyboard*) und Computer-Maus folgendermaßen steuern:

- Hält man im *Scene View*-Bereich die rechte Maustaste, so wie es auch in anderen Spielen üblich ist, gedrückt, kann man mit der Computer-Maus und den gedrückten *Keyboard*-Tasten *W* (vorwärts), *S* (rückwärts), *A* (links), *D* (rechts) in der „Scene“ mit langsamen Vor- und Zurückbewegen „herumfliegen“.
- Durch Rotation der „Kamera“ im Unity 3D-Editor mit der Computer-Maus kann der Anwender auch die Richtung des *Game Objects* bestimmen. Zum Drehen der *Game Object*s kann man auch die rechte Maustaste geklickt halten und die Computer-Maus hin- und zurückbewegen.
- Die *Keyboard*-Tasten *E* und *Q* erlauben es, das *Game Object* senkrecht hoch und hinunter fliegen zu lassen. Die markierten *Game Objects* werden dann auch in der *Hierarchy* markiert.
- Das Zoomen erfolgt mit dem Mausrad. Wenn man das Mausrad gedrückt hält, dann erscheint in der Anwendung ein „Händchen“ und man kann das *Game Object* durch Bewegung der Computer-Maus verschieben.

Unity 3D 2018 bietet unter der Unity-*Toolbar* (siehe dazu Abb. 56) auch unterschiedliche Werkzeuge, mit denen man ein ausgewähltes *Game Object* wunschgemäß anpassen oder verändern kann.

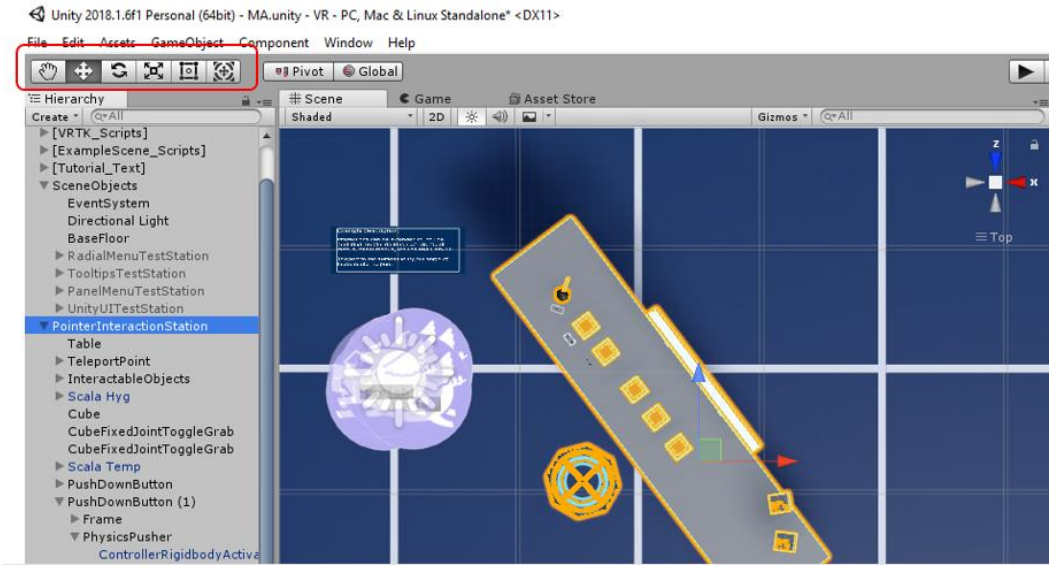


Abb. 56: Transform-Tools in Unity 3D 2018-Editors, Quelle: Eigene Darstellung.

Hand-Tool:

Mit diesem Werkzeug kann man die Perspektive in der Scene View verschieben. Beim Aktivieren wird anstelle dem Mauszeiger ein Hand-Symbol in Unity 3D 2018 angezeigt.

Translate-Tool:

Das Kreuz aus Pfeilen aktiviert das Verschiebe-Werkzeug. Ist dieses Werkzeug aktiviert, werden an dem ausgewählten *Game Object* in der *Scene View* Pfeile angezeigt, mit denen man das *Game Object* dann verschieben kann. Objekte können in der *Scene* verändert werden, auch während der Spieleausführung. Beim Beenden wird die Anpassung jedoch wieder zurückgesetzt.

Rotate-Tool:

Die beiden Pfeile aktivieren das Rotations-Werkzeug. Bei diesem Werkzeug werden in der *Scene View* mehrere Kreise angezeigt, mit denen man das ausgewählt *Game Object* drehen kann.

Scale-Tool:

Die Schaltfläche mit dem Quadrat und den vier Pfeilen aktiviert das Skalierungs-Werkzeug. Dieses Werkzeug erlaubt es, *Game Objects* in der *Scene View* zu vergrößern oder zu verkleinern.

Rect-Tool:

Diese Schaltfläche ist ein Werkzeug, das vor allem für *User Interfaces* (z.B. Menüs) verwendet wird. Es erlaubt Bilder, Schaltflächen und andere Elemente auf einer 2D-Ebene zu verschieben und in der Größe zu ändern. Dabei „docken“ die Elemente an allen passenden Kanten an. Dieses Werkzeug dient dazu, aufgeräumte und einheitlich aussehende *User Interfaces* zu erstellen.

Project Browser:

Ein weiterer Bereich im Unity 3D-Editor ist der *Project Browser*. Im Project Browser findet der Anwender alle Assets die zum Projekt gehören und kann nahezu jede Art von *Asset* auch neu anlegen.

Über dieses Menü kann man, wie auch in Abb. 57 zu sehen ist, weitere Ordner (*folder*) anlegen, um das Projekt besser zu sortieren.

Die *Shortcuts* (mit gelben *Icons* versehen) unterstützen bezüglich häufig verwendeter Suchanfragen. Mit dem Suchfeld im rechten oberen Teil des *Project Browsers* können Dateien und bestimmte *Assets* auch bequem gesucht und gefunden werden.

Die *Console* findet man in der Standardansicht als einen kleinen Reiter neben dem *Project Browser*. Der Anwender kann über die Reiter jederzeit zwischen *Console* und *Project Browser* hin und her wechseln.

In der *Console* werden hilfreiche Informationen sowie Warnungen und Fehlermeldungen angezeigt, wenn etwas nicht funktioniert, wie es soll. *Scripts* können hier jederzeit sogenannte *Debug-Logs* hineinschreiben. Diese Ausgaben dienen dazu, Fehler in der Anwendung (*Bugs*) zu finden. Diese *Debug*-Informationen können Aufschluss darauf liefern, wann und warum ein Fehler aufgetreten ist.[60]

[60] Vgl. Korgel (2018), S. 44 ff.

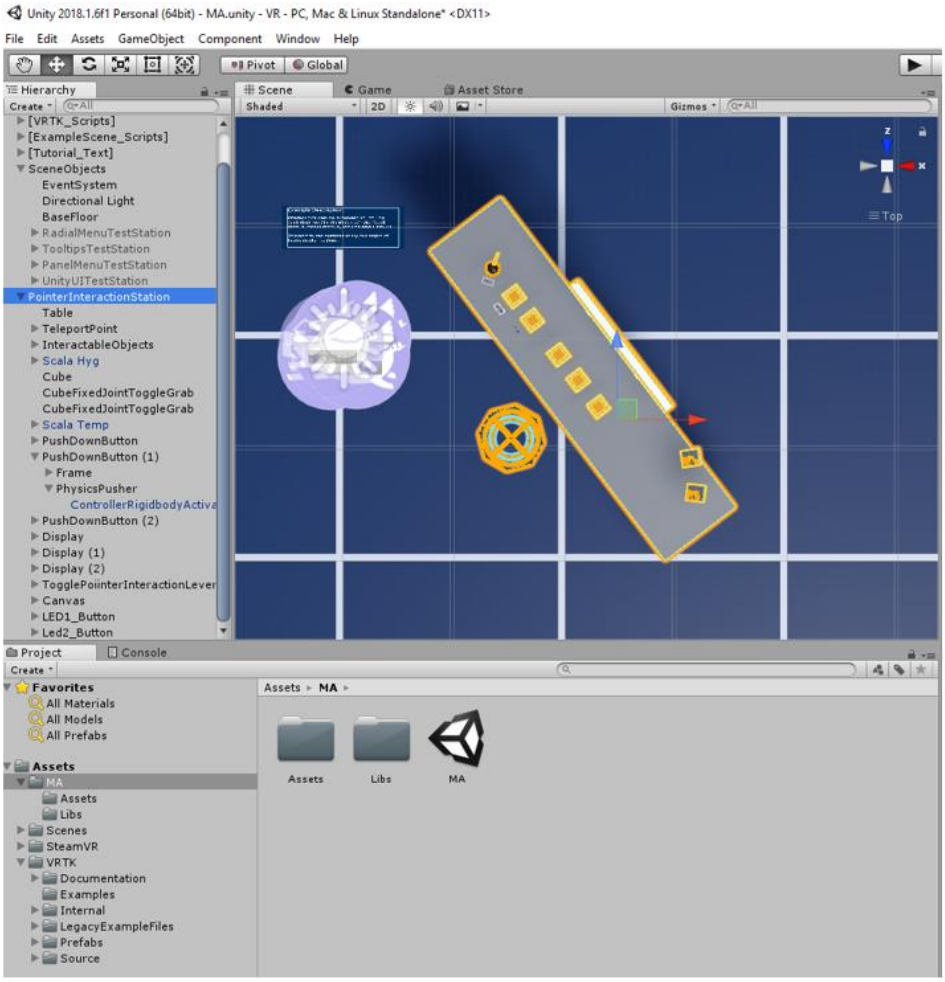


Abb. 57: Project Browser in Unity 3D 2018-Editors, Quelle: Eigene Darstellung.

11.2 Die Virtual Reality-Entwicklungswerkzeuge (SDK)

Jede Virtual Reality-Brille besitzt ihre eigenen Entwicklungswerkzeuge (SDK), die trotz der Unity-internen Virtual Reality Unterstützung benötigt werden.

SKD steht für *Software-Development-Kits*, die meistens von den Herstellern einer Hard- oder Software für externe Entwickler zur Verfügung gestellt werden, z.B. das Oculus SDK zum Entwickeln für Oculus Virtual Reality-Brillen (siehe dazu auch Abb. 58).

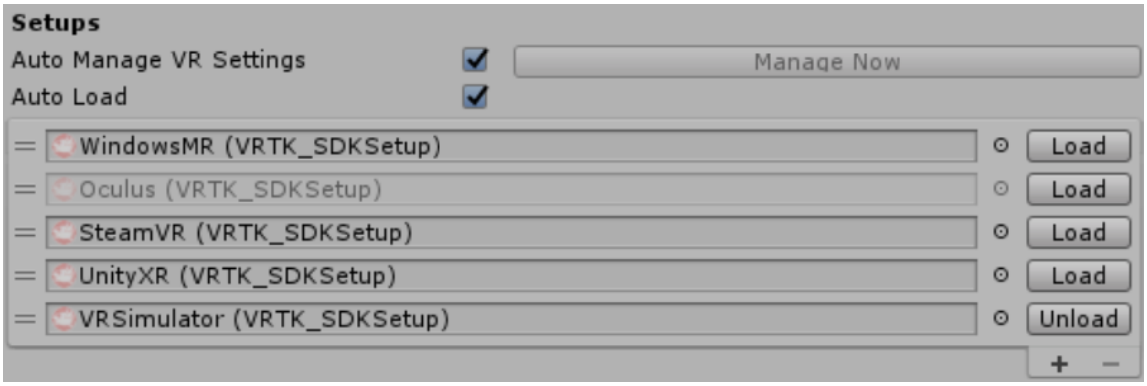


Abb. 58: Die Virtual Reality-SDKs unter Unity 3D 2018, Quelle: Eigene Darstellung.

11.3 SteamVR (für die HTC Vive Virtual Reality-Brille)

Die Entwicklungswerkzeuge für SteamVR benötigt man, wenn man eine Virtual Reality-Anwendung (z.B. einen virtuellen Messtisch) für eine SteamVR kompatible Brille wie die HTC Vive Virtual Reality-Brille entwickeln möchte.

Die Schnittstelle von SteamVR wird auch *OpenVR* genannt, da sie auch andere Virtual Reality-Brillen, wie z.B. die Oculus Rift unterstützt.

In den folgenden Abschnitten wird auf wesentliche Features im Zusammenhang mit dem Arbeiten mit SteamVR wie folgt näher eingegangen:[61]

Das SteamVR-SDK ist so geschrieben, dass die meiste Logik in der *Runtime* (der SteamVR Anwendung) liegt. Man implementiert lediglich die Schnittstelle zu SteamVR, die SteamVR-Anwendung enthält dann die Logik, die benötigt wird, die VR-Brille anzusprechen.

SteamVR unterscheidet zwischen *Tracking-Space* und *Play-Area*. Der *Tracking-Space* ist der komplette zur Verfügung stehende Bereich, an dessen Grenzen das *Chaperone* eingeblendet wird, damit man nicht gegen reale Gegenstände stößt. *Chaperone* heißt das Sicherheitssystem von SteamVR, welches in der virtuellen Welt die Grenzen des *Tracking-Space* anzeigt.

Der *Tracking-Space* muss nicht exakt rechteckig sein, sondern kann exakt dem zur Verfügung stehenden Raum entsprechen. Für die Entwicklung einer virtuellen Umgebung ist es allerdings sehr umständlich, beliebige *Tracking-Space* Formen berücksichtigen zu müssen. Daher gibt es zusätzlich noch die *Play-Area*, welches das größtmögliche Rechteck darstellt, das in den *Tracking-Space* passt. Bei der Entwicklung einer Anwendung oder eines Spieles kann der Anwender jeden Punkt innerhalb dieses rechteckigen Bereiches erreichen.

Das *Controller-Manager-Component* aktiviert und deaktiviert die *Game-Objects-Controller left* und *right*, wenn ein Controller mit SteamVR verbunden beziehungsweise die Verbindung getrennt wird. Wenn man einen Controller einschaltet, wird von SteamVR basierend auf ihrer Position (links oder rechts von der VR-Brille) bestimmt, ob sich der Controller gerade in der linken oder rechten Hand des Anwenders befindet. Aktiviert man die Option *„Assign All Before Identified“*, wird anstelle der Position die Reihenfolge des Einschaltens verwendet, um den Controller einer Rolle zuzuordnen. Der erste Controller, der eingeschaltet wird, ist dann der rechte und der zweite der linke.

Das *Tracked-Object-Component* ist dafür zuständig, dass sich *Game-Objects* synchron zu einem getrackten Gegenstand in der realen Welt (z.B. VR-Brille, Controller) bewegen.

Das *Render-Model-Component* lädt automatisch ein Modell, das zu der Hardware des jeweiligen getrackten Gegenstands mit dem angegebenen Index passt. Der Index bestimmt, zu welchem getrackten Objekt das *Game Object* gehört (*HMD* ist die VR-Brille und *Device* 1 und *Device* 2 für die beiden *Controller*). Wird der Index eines *Controllers* angegeben und der Anwender verwendet z.B. HTC Vive Controller, werden Vive Controller in der virtuellen Welt angezeigt.

[61] Vgl. Korgel (2018), S. 368 ff.

11.4 Die Erstellung der einzelnen Skalen für das Hygrometer und das Thermometer und Einbindung in Unity 3D 2018

Die beweglichen Skalen für das analoge Hygrometer und das analoge Thermometer für Unity 3D 2018 werden in „Blender“ (Version 2.79) erstellt (siehe dazu auch Abb. 59).

Diese Software wurde von der Blender Foundation 1998 entwickelt und ist eine freie 3D-Grafiksuite, mit welcher sich Körper einfach erstellen, modellieren, texturieren und animieren lassen. Seine dynamische Benutzeroberfläche passt sich der jeweiligen Aufgabe an. Als Programmiersprechen werden C und C++ verwendet. Python dient als Skriptsprache. Die aktuelle Version (*release* 21. November 2019) ist V2.81.

Blender kann die erzeugten Szenen mittels verschiedener *Renderengines* zu einem finalen *Game Object* für die weitere Verwendung in Unity 3D 2018 ausgeben.

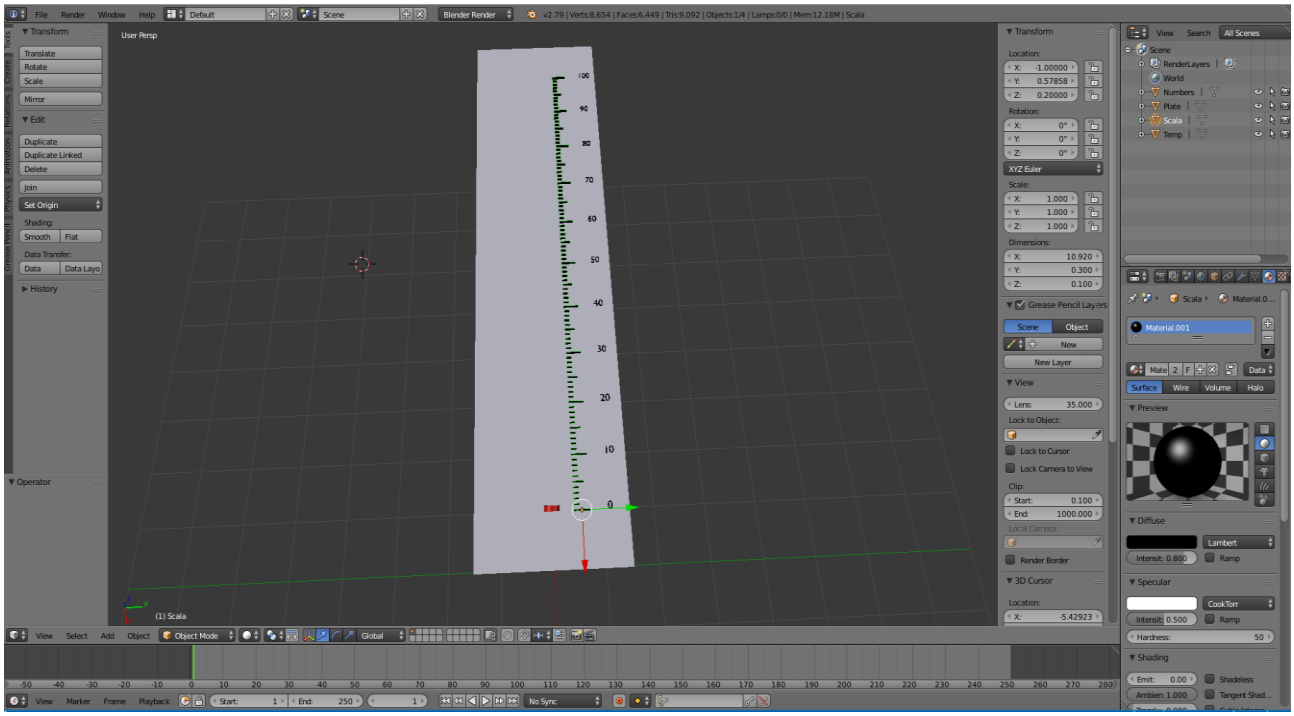


Abb. 59: Bewegliche Skalen des analogen Thermometers und des analogen Hygrometers, Quelle: Eigene Darstellung.

Das finale analoge Thermometer und das finale analoge Hygrometer werden anschließend als *Game Objects* in Unity 3D 2018 eingebunden. Zusätzlich müssen noch ein paar Codezeilen (*script*) geschrieben werden. Sobald das *script* im Projektordner abgelegt wird, startet Unity 3D 2018 automatisch den *Compiler*, der das *script* dann übersetzt.

Sollte ein *script* einen Fehler enthalten, wird ein Compiler-Fehler in der *Console* angezeigt. Unity startet grundsätzlich jeden Kompilierungsvorgang selbständig und automatisch jedes Mal, wenn vom Anwender eine Änderung am *script* vorgenommen wurde.[62]

[62] Vgl. Korgel (2017), S. 151.

11.5 Programmerstellung in Unity 3D 2018

Abb. 60: Virtual Reality Toolkit für die Unity Engine, Quelle: VRTK (2018), Online-Quelle [03.06.2019].

Virtual Reality Toolkit (VRTK) ist eine Sammlung nützlicher Skripte und Konzepte, mit deren Hilfe Virtual Reality-Lösungen in Unity 3D 2018 erstellt werden können. Es bietet Vorlagen und Lösungen für die Teleportation im virtuellen Raum, Interaktionen, wie das Berühren, das Greifen und das Verwenden von virtuellen Objekten, die Interaktion mit Unity 3D-Oberflächenelementen durch Zeigen oder Berühren, 2D- und 3D-Steuerelemente wie *Buttons*, Hebel, Türen, Schubladen usw. VRTK hat es sich dabei zum Ziel gesetzt, eine Community von Anwendern aus der ganzen Welt zusammenzubringen, um Lösungen für ungelöste Anforderungen und Herausforderungen in Bezug auf Virtual Reality zu schaffen und somit zur Weiterentwicklung dieser Technologie beizutragen.[63]

11.5.1 Softwarescript in Unity 3D 2018

Ein *Script* ist eine Textdatei, die Anweisungen in C# enthält. Bei der Ausführung wird dieses in Maschinensprache übersetzt (kompiliert), welche der Rechner schneller interpretieren kann. Das Programm, das das *Script* übersetzt nennt sich *Compiler*. Sollte ein *Script* einen Fehler enthalten, findet man den *Compiler*-Fehler in der *Console* in „Unity“. Neben C# unterstützt Unity 3D auch die Programmiersprache *JavaScript*.

C# ist eine objektorientierte Programmiersprache, die von Microsoft entwickelt wurde. In der folgenden Tabelle ist das Script für die virtuelle Steuerung der Arduino und LED-Boards dargestellt.

Softwareprogramm

```
using System.Collections;
using System.Collections.Generic;
using System.IO.Ports;
using System.Threading;
using UnityEngine;
using UnityEngine.UI;
using EasyModbus;
using VRTK.Controllables.PhysicsBased;


public class Lab_Communication : MonoBehaviour

{
    public GameObject test;
    public GameObject button_serial_object;
    public GameObject button_modbus_object;
    public  GameObject scala_temp_object;
    public  GameObject scala_hyg_object;

    public GameObject led_switch_object;

    UnityEngine.UI.Dropdown dropdwon_;
```

[63] Vgl. VRTK (2018), Online-Quelle [03.06.2019].

```
    UnityEngine.UI.Button button_serial_;
    UnityEngine.UI.Button button_modbus_;
    // Use this for initialization
    static ModbusClient modbusClient = new ModbusClient("127.0.0.1", 502);
    static bool modbus_connected = false;
    string part_name_ ="";
    string message;
    static bool _continue = true;
    static SerialPort _serialPort;
    System.StringComparer stringComparer = System.StringComparer.OrdinalIgnoreCase;
    Thread readThread = new Thread(Read);

    static float temp_current_ = 0.0f;
    static float hyg_current_ = 0.0f;

    float smooth_scale_temp = 0.0f;
    float smooth_scale_hyg = 0.0f;
    float smooth_value = 0.05f;

    ~Lab_Communication()
    {
        _serialPort.Close();
    }
    void Start () {
        dropdwon_ = test.GetComponent<UnityEngine.UI.Dropdown>();
        button_serial_ = button_serial_object.GetComponent<UnityEngine.UI.Button>();
        button_modbus_ = button_modbus_object.GetComponent<UnityEngine.UI.Button>();

        dropdwon_.onValueChanged.AddListener(delegate {
            DropdownValueChanged(dropdwon_);
        });

        button_modbus_.onClick.AddListener(connectToModbus);
        button_serial_.onClick.AddListener(connectToComport);

        dropdwon_.options.Clear();

        string[] ports = SerialPort.GetPortNames();

        Debug.Log("The following serial ports were found:");

        //Display each port name to the console.
        foreach (string port in ports)
        {
            Dropdown.OptionData m_NewData = new Dropdown.OptionData();
            m_NewData.text = port;
            dropdwon_.options.Add(m_NewData);
            Debug.Log(port);
        }

        //serialPort = new SerialPort(@"\\.\" + "COM16",115200);


       //serialPort.Open();

       //readThread.Start();
    }

    //Update is called once per frame
    void Update () {
        //string message = _serialPort.ReadLine();
        //Debug.Log(message);

        if (smooth_scale_temp != temp_current_)
```

```
            smooth_scale_temp = smooth_scale_temp > temp_current_ ? smooth_scale_temp -
smooth_value : smooth_scale_temp + smooth_value;

        if (smooth_scale_hyg != hyg_current_)
            smooth_scale_hyg = smooth_scale_hyg > hyg_current_ ? smooth_scale_hyg -
smooth_value : smooth_scale_hyg + smooth_value;

        scala_temp_object.transform.localScale = new Vector3(1 + smooth_scale_temp *
100.0f, 100.0f, 100.0f);
        scala_hyg_object.transform.localScale = new Vector3(1 + smooth_scale_hyg *
100.0f, 100.0f, 100.0f);
    }

    public  float getTemp()
    {
        return temp_current_;
    }

    public  float getHyg()
    {
        return hyg_current_;
    }

    public void  switchLed_1_On()
    {
        if(_serialPort != null)
            _serialPort.WriteLine("1");
    }

    public void switchLed_2_On()
    {
        if (_serialPort != null)
            _serialPort.WriteLine("2");
    }

    //Events
    void DropdownValueChanged(Dropdown change)
    {
        part_name_ = dropdwon_.options[change.value].text;
        Debug.Log("Changed to " +part_name_);
    }

    public void connectToComport()
    {

        //if (readThread.IsAlive)
        /     readThread.Abort();

        if(_serialPort != null)
            if (_serialPort.IsOpen)
                _serialPort.Close();

        if (part_name_ == "")
            return;


        _serialPort = new SerialPort(@"\\.\" + part_name_, 115200);

        _serialPort.Open();

        readThread.Start();

        ColorBlock cb = button_serial_.colors;
        cb.normalColor = Color.green;
```

```
        button_serial_.colors = cb;
    }

    public void connectToModbus()
    {
        modbusClient.Connect();

        modbus_connected = true;
        ColorBlock cb = button_modbus_.colors;
        cb.normalColor = Color.green;
        button_modbus_.colors = cb;
    }

    public static void writeToModbus(int reg,float value)
    //reg:  0: temp and 1:hygrometer
    {
        if (modbus_connected)
        {
            Debug.Log("Write to Modbus");
            modbusClient.WriteSingleRegister(reg, (int)value);
        }
    }

    //Threads

    public static void Read()
    {
        Debug.Log("Serial Read start");
        while (_continue)
        {
            try
            {
                string message = _serialPort.ReadLine();
                Debug.Log(message);

                switch (message[0])
                {
                    case 'T':
                        temp_current_ = float.Parse(message.Remove(0, 1));
                        writeToModbus(0, temp_current_);
                        //scala_temp_object.transform.localScale = new
Vector3(1+temp_current_ * 100.0f, 100.0f, 100.0f);
                        break;
                    case 'H':
                        hyg_current_ = float.Parse(message.Remove(0, 1));
                        writeToModbus(1, hyg_current_);
                       // scala_hyg_object.transform.localScale = new
Vector3(1+hyg_current_ * 100.0f, 100.0f, 100.0f);
                        break;

                }
            }
            catch (System.TimeoutException) {
                Debug.Log("Timeout");
            }
        }
    }

}
```

Tabelle 11: C#-Script, geschrieben in Microsoft Visual Studio, Quelle: Eigene Darstellung.

11.5.2 Verschiedene Datentypen in C#

Einige Datentypen können direkt mit C#-Schlüsselwörtern in Microsoft Visual Studio beschrieben werden (siehe dazu auch Abb. 61). Diese Schlüsselwörter sind dabei nur Aliasnamen für entsprechende .NET-Framework-Datentypen. Sie bilden den Kern dieser Sprache und stehen für eine andere Verwendung (z.B. für Variablennamen) nicht zur Verfügung.

In C# gibt es beispielsweise 9 *integral-types*:

- *byte,*
- *sbyte,*
- *short,*
- *ushort,*
- *int,*
- *uint,*
- *long* und
- ulong usw.

2 *floating point types*: *float* und *double*, einen *bool-type*, einen *char-type*, einen *struct-type* usw.

Bool ist einer der einfachsten Datentypen. Er kann nur einen von zwei Werten annehmen nämlich true or false. Dieser Datentyp wird bei logischen Operationen, wie z.B. der if-Anweisung verwendet.

Int steht für *Integer*. Darunter versteht man einen Datentyp zum Speichern von Variablen ohne Dezimalstellen. Er ist der Häufigste verwendete Datentyp.

String wird zum Speichern von Texten verwendet. *Strings* können in C# nicht abgeändert werden, nachdem sie erzeugt wurden. Der Original-*String* bleibt dabei unverändert und stattdessen wird ein neuer *String* erstellt und ausgegeben.

Char wird zum Speichern eines einzelnen Zeichens benutzt.

Float ist einer der Datentypen, der zum Speichern von Zahlen, die auch Dezimalstellen haben, benutzt wird.[64]

[64] Vgl. C# Tutorial (2004), Online-Quelle [30.11.2019].

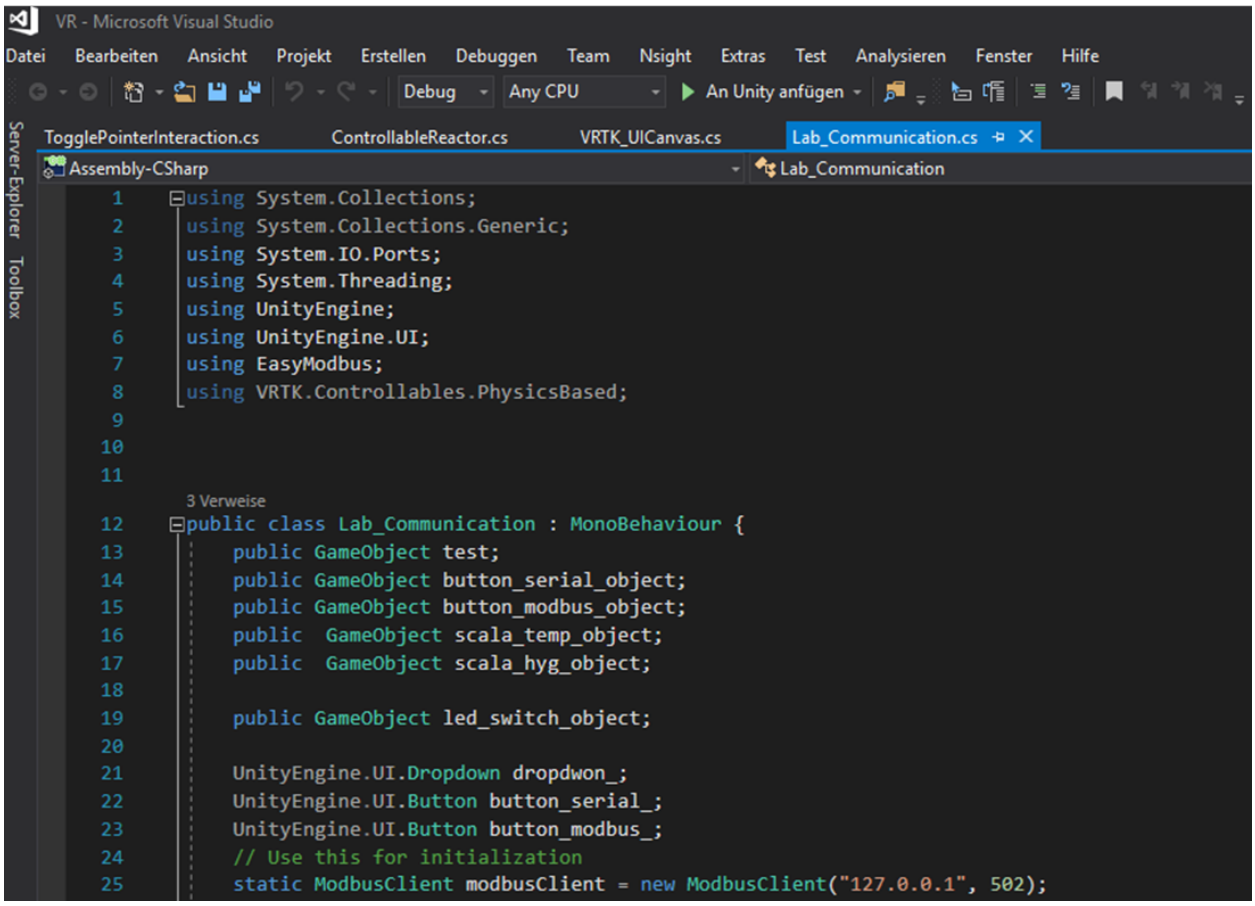


Abb. 61: Auszug aus dem C#-script in Unity 3D 2018, Quelle: Eigene Darstellung.

11.6 Modellerstellung eines virtuellen Messtisches in Unity 3D 2018

Gemäß dem Blackbox-Modell kann ein erster Software-Entwurf des Messtisches, wie in der nachfolgenden Abbildung dargestellt für die virtuelle Umgebung in Unity 3D 2018 erstellt und visualisiert werden.

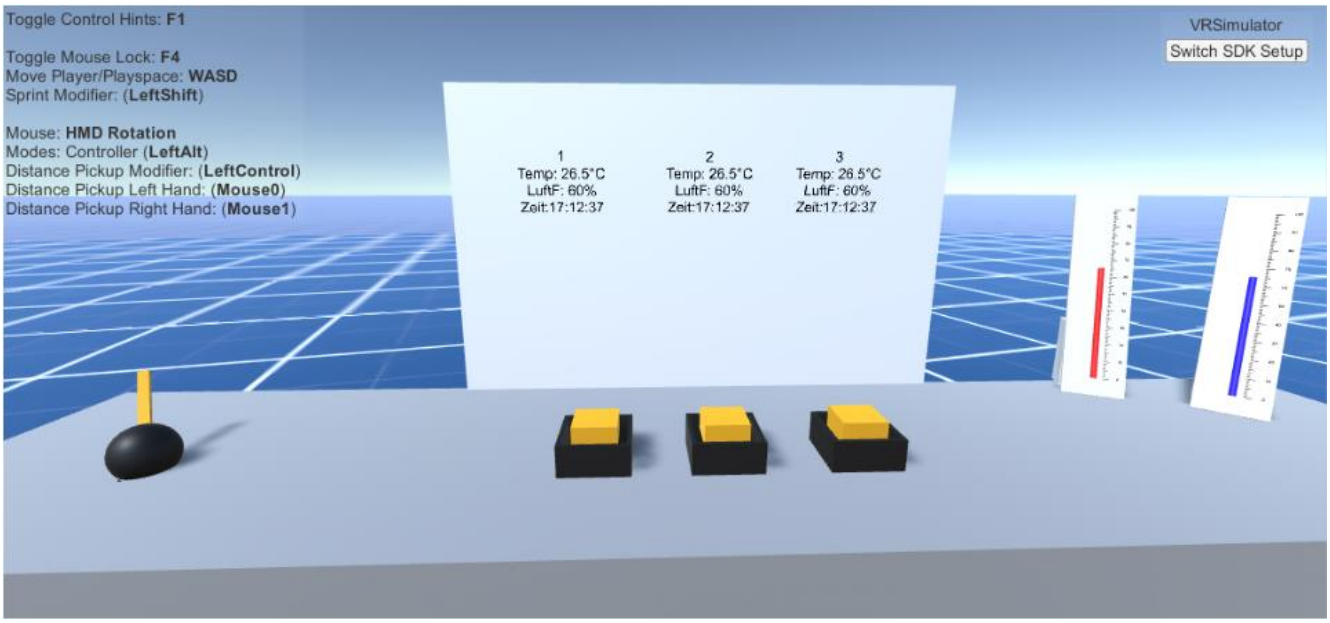


Abb. 62: Messtisch für die virtuelle Umgebung, Quelle: Eigene Darstellung.

In Abb. 62 ist ein virtueller Messtisch mit 3 Tasten dargestellt. Wenn jeweils einer der gelben Tasten virtuell (z.B. über die Computer-Maus oder die Virtual Reality-Controller) betätigt werden, werden die Temperatur, die relative Luftfeuchtigkeit, sowie die Zeit des jeweiligen Abrufes der Messwerte am *Display* angezeigt und bis zum erneuten Betätigen des jeweiligen Tasters werden die Daten auf dem virtuellen Display gespeichert.

Auf diese Weise können die Werte von jeweils drei unterschiedlichen Messungen dargestellt und eine mögliche Veränderung der Messwerte über die Zeit verglichen werden. Natürlich darf für den laufenden Datenaustausch die Verbindung zum Messsensor über das Arduino kompatible Board nicht unterbrochen werden.

Die Boxen mit den gelben Tasten (*Taster*) sowie das *Display* können am virtuellen Tisch nicht verschoben werden. Der Messtisch selbst ist in der Spieleumgebung so räumlich auszurichten und zu platzieren (durch Drehen, Positionieren und Ausrichten anhand des Koordinatensystems in der Software), dass dieser mit den Bediengeräten immer leicht detektiert werden kann und keine anderen virtuellen Elemente aus der VRTK-Vorlagen- und Beispielsammlung in Unity 3D 2018 den direkten Sichtkontakt auf die jeweiligen Elemente in der virtuellen Umgebung versperren oder behindern. Hier sollte also von Anfang an auf eine saubere und gut durchdachte Struktur Wert gelegt werden um spätere Kollisionen und Modifizierungen zu vermeiden.

Daher sollte bereits in der Phase der Modellerstellung das Layout und die benötigten Komponenten bzw. virtuellen Elemente gut und übersichtlich bestimmt und ausgewählt werden. Denn wenn in weiterer Folge in der Realen Welt die Virtual Reality-Antennen (Virtual Reality-Basis-Stationen) auf Grund von nicht gut durchdacht platzierten virtuellen Elementen im Koordinatensystem den direkten Sichtkontakt zu den Virtual

Reality-Controllern verlieren (z.B. wenn man die virtuellen Elemente über die VR-Brille anwählen möchte, sich dabei im realen Raum um die eigene Körperachse dreht und den Basis-Stationen, wenn auch nur zeitweise, „den Rücken zukehrt“, d.h. außerhalb des Sichtfensters der VR-Brillen zu den Basis-Stationen gelangt), wird die Verbindung zwischen Antenne und der VR-Hardware sofort unterbrochen und es ist ein erneutes manuelles Koppeln der Hardwarekomponenten mit der Software notwendig. Dies ist mit aufgesetzter Virtual Reality-Brille allerdings nicht möglich, da das Koppeln immer manuell über die Benutzerführung in Unity 3D 2018 erfolgen muss.

Mit Hilfe des Schiebers am virtuellen Messtisch können zusätzlich LEDs nach unterschiedlichen Algorithmen angesteuert werden. Diese können auf diese Weise auf dem entwickelten LED-Board nach klar definierten Regeln zum Leuchten gebracht werden, sobald man den virtuellen Hebel des Schiebers nach vorne schiebt, eine Mittelposition einnimmt oder wieder zurückbewegt.

Prinzipiell können in der Spieleengine Unity 3D 2018 bereits vorhandene vorprogrammierte Beispiele aus der VRTK-Sammlung verwendet werden. Das *Eventscript* für die jeweilige Anwendung bzw. Anforderung muss vom Anwender lediglich noch angepasst werden.

Die Skalen für das analoge Thermometer und Hygrometers werden über die Software „Blender“ (Version 2.79) erstellt und danach wieder in die Spieleumgebung von Unity 3D 2018 integriert. Beide Elemente sind an einen, ebenfalls verschiebbaren Sockel angelehnt und können mit der Computer-Maus bzw. den Virtual Reality-Controllern aufgenommen und herangezoomt werden, um die aktuellen, analog dargestellten realen Messwerte auf der Skala besser ablesen zu können.

Mit einer Abtastrate von ca. 2 Sekunden können somit Veränderungen bezüglich der gemessenen Temperatur und der relativen Luftfeuchtigkeit über einen beweglichen Balken angezeigt und abgelesen werden. Diese Elemente (*Assets*) können beliebig im definierten Spielebereich (definierte virtuelle Raumgrenzen) platziert bzw. abgelegt werden.

11.7 Trouble-Shooting in der Umsetzung in Unity 3D 2018

Jedem, der schon einmal ein Programm erstellt bzw. ein bestehendes *skript* an die gestellten Anforderungen und Erwartungen angepasst hat, kennt das Problem, dass Fehlermeldungen im Zuge der Code-Erstellung oder Skript-Anpassung angezeigt werden oder dass Funktionen nicht ausgeführt werden können aber auch dass Hardwarekomponenten sich einfach nicht ansteuern lassen, unendlich viel Zeit und menschliche Energie verschlingen, um die Ursache zu ermitteln und eine Lösung dafür zu finden, um das jeweilige Probleme zufriedenstellend zu beheben.

Ein solches Problem ist es, den selbst konzipierte Prototyp eines LED-Boards (siehe dazu auch Kapitel 8) anzusteuern. Nachdem das Arduino kompatible Board von Windows trotz zahlreicher ergebnisloser Versuche nicht erkannt wird, muss sehr viel Zeit für die Lösungsfindung aufgewendet werden. Nach langer Internetrecherche kann in einem Internetforum ein Hinweis auf einen USB-Chipsatz CH340 eines chinesischen Anbieters gefunden werden. Da die Downloadseite für den Treiber für Windows allerdings nur in chinesischer Sprache also mit rein chinesischen Schriftzeichen verfügbar war und auch sämtliche Übersetzungsprogramme an ihre Grenzen stießen, war eine auf viel Vertrauen auf eine gute Quelle für den

Download dieses Treibers notwendig. Das Problem konnte mit diesem Treiber dann rasch und einfach behoben werden und das Skript konnte problemlos ausgeführt werden.

Nachfolgend sind noch zwei Abbildungen des fertigen virtuellen Messtisches in Unity 3D 2018 dargestellt.

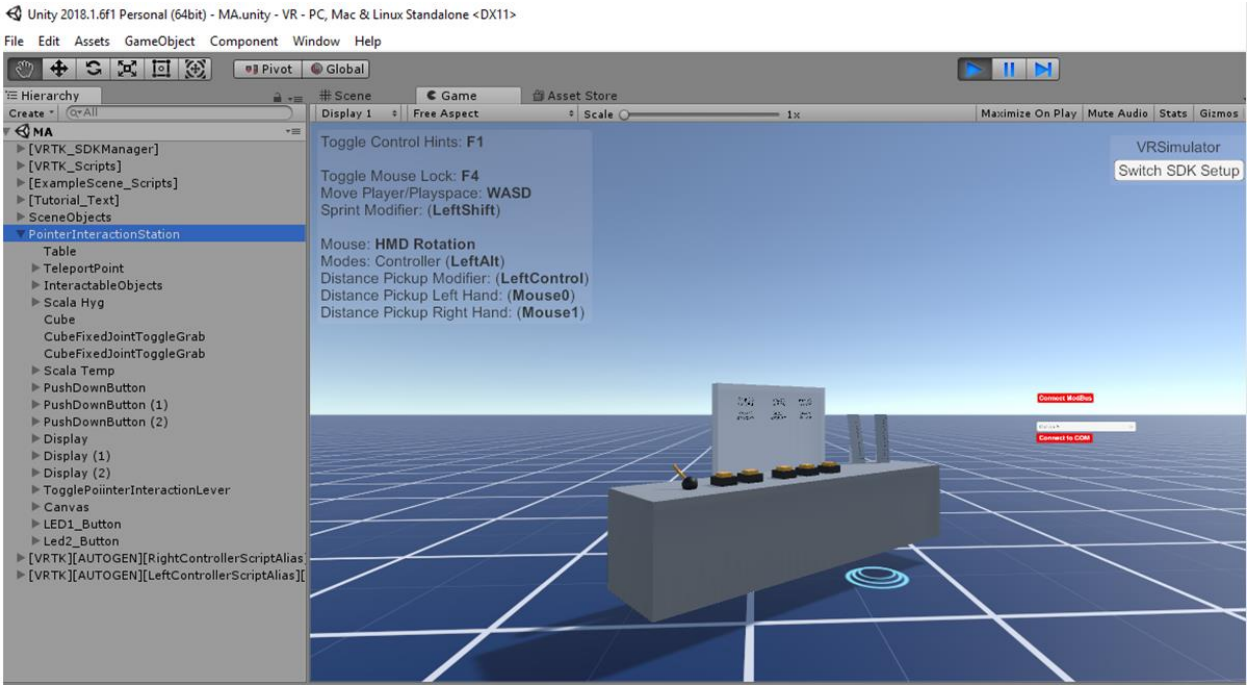


Abb. 63: Messtisch in der virtuellen Umgebung in Unity 3D 2018, Quelle: Eigene Darstellung.

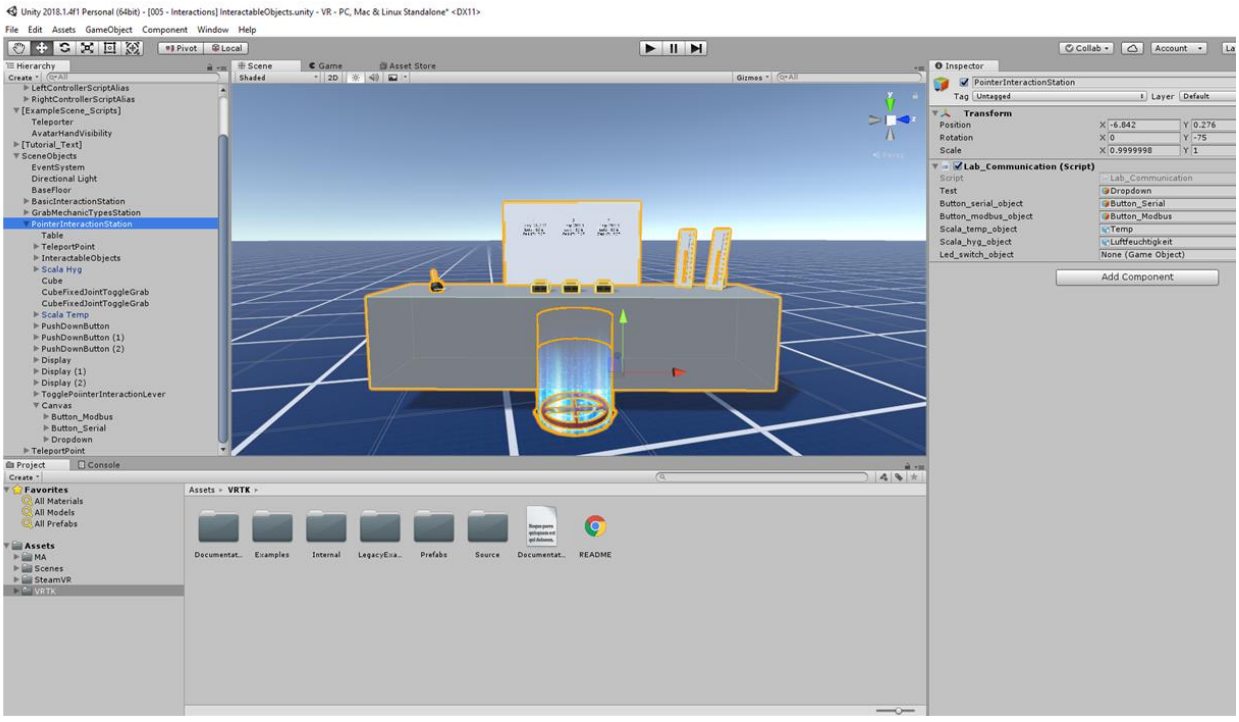


Abb. 64: Messtisch in der virtuellen Umgebung in Unity 3D 2018, Quelle: Eigene Darstellung.

11.8 Schnittstelle Modbus-TCP für Unity 3D 2018

Modbus-TCP ist ein *Client/Server*-Protokoll für den verbindungsorientierten gesicherten Austausch von Prozessdaten. Die Rollen des *Clients* und des *Servers* sind nicht fest zugeordnet. Jedes Gerät im Netzwerk kann beide Rollen spielen. Somit können Zugriffe auf Daten – lesend oder schreibend – flexibel den jeweiligen Aufgabenstellungen angepasst werden. Der *Server* bearbeitet eine Anfrage des *Clients* – den so genannten *Request* – und quittiert die Anfrage mit einer Erfolgsmeldung (*Response*), der gegebenenfalls angefragte Daten oder Statusinformationen mitgegeben werden, bzw. mit einer Fehlermeldung, die Informationen über die Fehlerursache enthält. Daten, die mit Modbus-TCP übertragen werden, können Bit- und Wortinformationen enthalten.[65]

Die Länge eines Datenblocks, der in einem Modbus-Telegramm übertragen werden kann, beträgt 252 Bytes. In diesem Telegramm sind die Geräteadresse (1 Byte), der Function Code (1 Byte) und die Datensicherung mit CRC (2 Bytes) enthalten. Mit dem Function Code wird festgelegt, welche Operation auf Grund eines *Requests* durch den Server ausgeführt werden soll. Als Folge hieraus lassen sich Modbus-*Requests* mit minimalem Aufwand von seriellen Übertragungsstrecken umsetzen. Die Art und Weise des Zugriffs auf Gerätedaten wird über Funktionscodes gesteuert.

In der Ausführung dieser Masterarbeit wurde entschieden eine Applikation, nämlich den Modbus Server Simulator zu verwenden. Der Modbus Server Simulator hilft Programmentwicklern bei der Verwendung des Modbus-Protokolls. Alle Modbus-Daten werden im Serversimulator „Easy Modbus Server Simulator“ angezeigt und ermöglichen das Debuggen von Client-Applikationen. Diese Serversimulation eines Modbus TCP-Clients wurde in Visual Studio 2012 geschrieben und ermöglicht es dem Anwender, dass Werte des Datentyps „*Float*“ (32 Bit, für Nachkommastellen) gelesen und geschrieben werden können.

Die folgenden Funktionscodes werden unterstützt:

- *Read Coils* (FC 01)
- *Read Discrete Inputs* (FC 02)
- *Read Holding Registers* (FC 03)
- *Read Input Registers* (FC 04)
- *Write Single Coil* (FC 05)
- *Write Single Register* (FC 06)
- *Write Multiple Coils* (FC 15)
- *Write Multiple Registers* (FC 16)
- *Read/Write Multiple Registers* (FC 23)

Der Serversimulator unterstützt Modbus-TCP, Modbus UDP und Modbus RTU in der .NET-Version. Die JAVA-Version unterstützt ebenfalls Modbus-TCP. Diese Serversimulation eines Modbus-TCP-Clients wurde in Visual Studio 2012 geschrieben (siehe dazu auch Abb. 65) und ermöglicht es dem Anwender, Werte des Datentyps „*Float*“ (32 Bit) zu lesen und zurückzuschreiben.

[65] Vgl. Feldbusse.de (2019), Online-Quelle [10.06.2019].

Die Einbindung von Modbus-TCP erfolgt in Unity 3D über „Anwendung“ in Microsoft Visual Studio (siehe dazu auch Abb. 65).

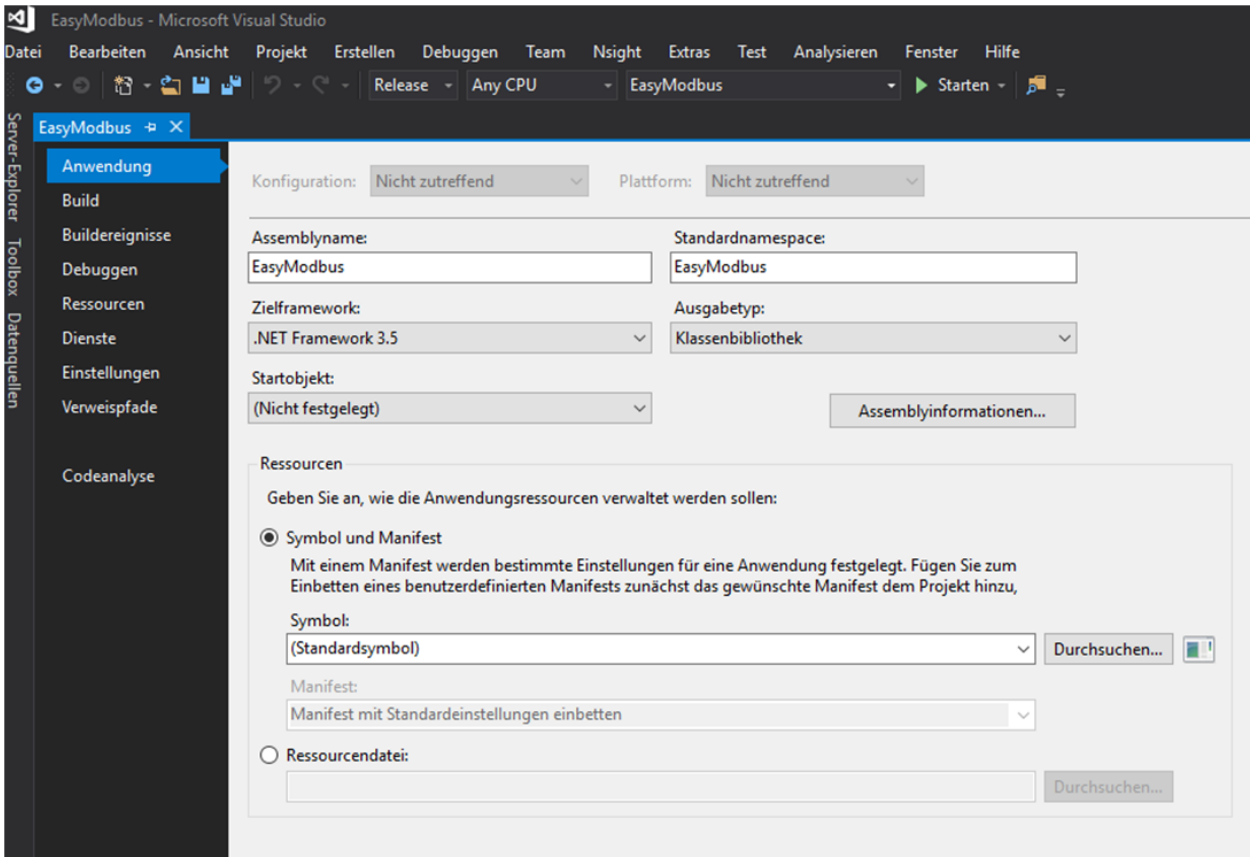


Abb. 65: Einbindung der Anwendung EasyModbus in Unity, Quelle: Eigene Darstellung.

Der Download der *.exe*-Datei steht unter http://easymodbustcp.net/en/ kostenlos zur Verfügung. Die *.exe*-Datei ist vor der Anwendungsausführung in Unity 3D zu aktivieren (siehe dazu auch Abb. 66)

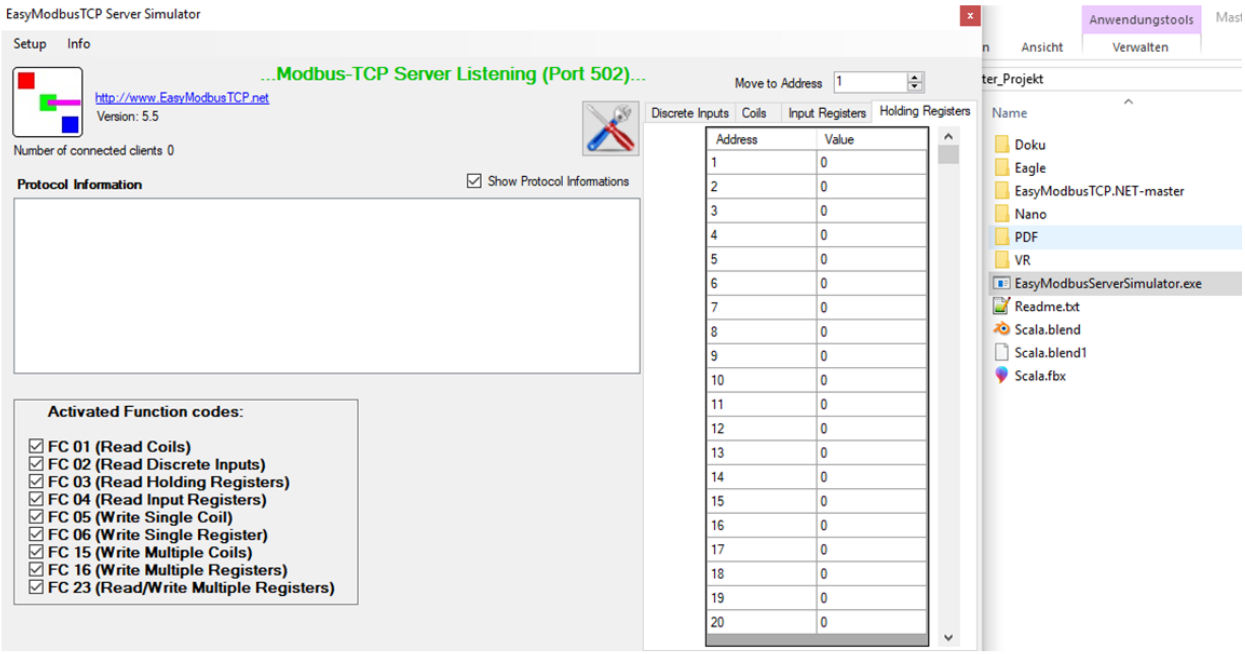


Abb. 66: Easy Modbus Server Simulator, Quelle: Eigene Darstellung.

Nach Auswahl des richtigen COM-Ports (*Connect to COM*) und Verbindung mit Modbus-TCP (*Connect ModBus*) in Unity 3D 2018 werden im Reiter „*Holding Registers*“ auf der Adresse 1 die aktuelle Raumtemperatur (diese betrug zum Zeitpunkt des Tests 23 ° Celsius) und auf der Adresse 2 die relative Luftfeuchtigkeit (60 %) angezeigt (siehe dazu Abb. 67 und Abb. 68).

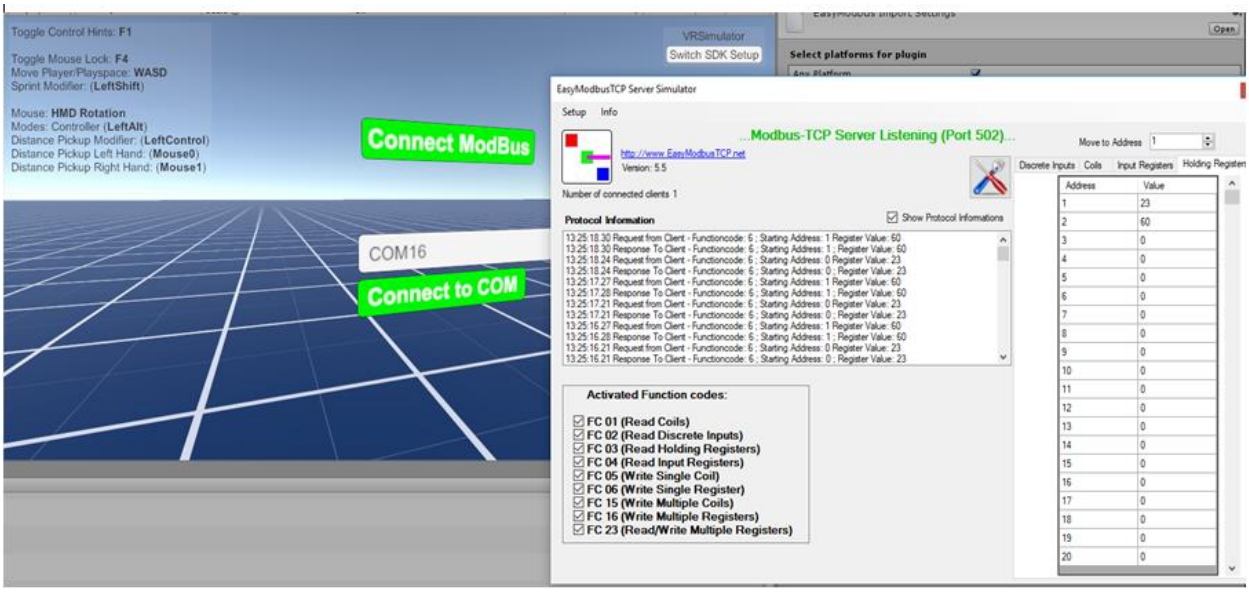


Abb. 67: Easy Modbus Server Simulator – Schnittstelle in Unity, Quelle: Eigene Darstellung.

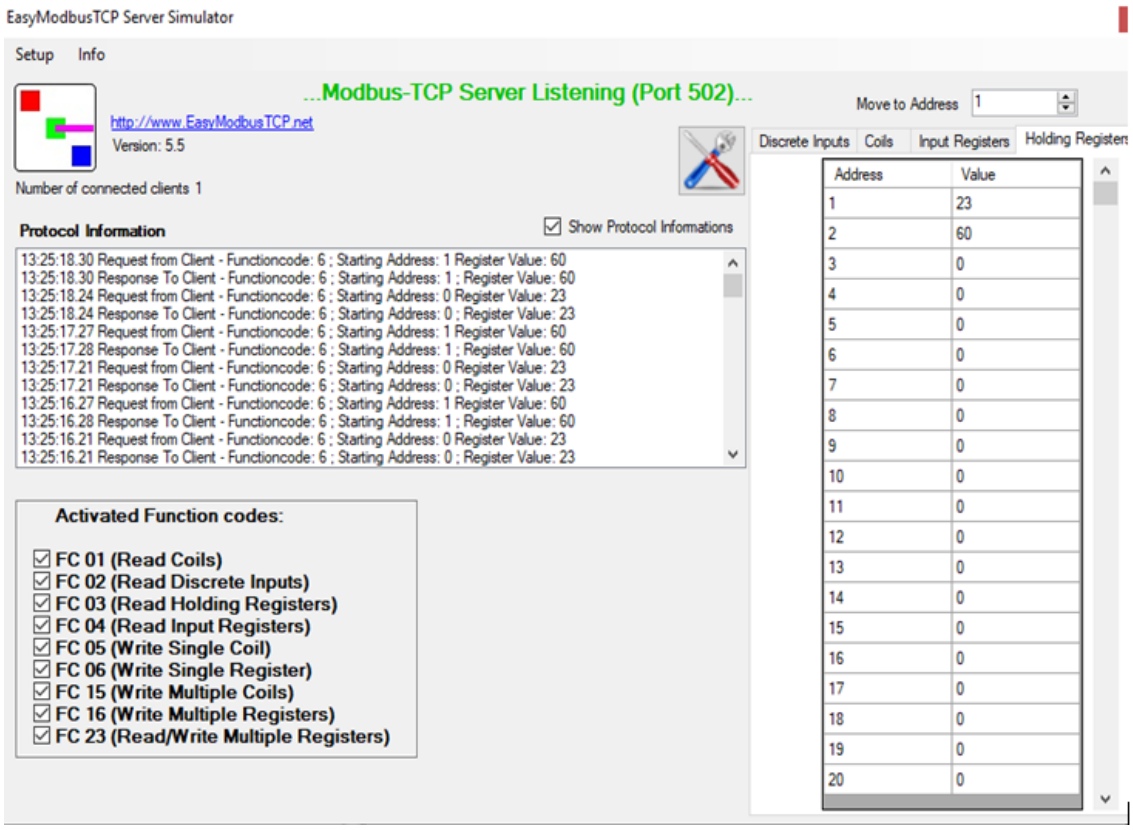


Abb. 68: Easy Modbus Server Simulator – Holding Registers, Quelle: Eigene Darstellung.

12 ERKENNTNISSE, ERGEBNISSE UND AUSBLICK

12.1 Erkenntnisse

Sowohl der Bereich der Virtual Reality als auch Augmented Reality haben sich in den letzten Jahren technologisch grundlegend verändert. Die Gründe dafür sind die enorme Weiterentwicklung in der Hardware wie die Verfügbarkeit von leistungsstarken Prozessoren, guten Grafikkarten und der Ausbau der Internetverbindungen und Bandbreiten, die den Anwendern zur Verfügung stehen. Vor allem werden diese Technologien auch immer stärker im industriellen Umfeld eingesetzt und professionell genutzt. Die VR-Hardware ist auch für den breiten Massenmarkt in einer schon recht guten Qualität auch leistbar geworden.

Das Konzipieren einer kleinen Schaltung sowie eines Prototypen ist für einen Nicht-Elektroniker eine spannende Herausforderung, bei der man viele unterschiedliche Bereiche der Automatisierungstechnik (z.B. Sensorik, Programmierung, usw.) abdecken kann. Die Verwendung von nicht originalen Hardwarekomponenten kann bei der Umsetzung von Anwendungen jedoch sehr schnell einen großen Zeitaufwand verursachen, um diese in eine Software-Umgebung einzubetten, da Treiberprobleme oder Schnittstellenprobleme auftreten können oder die Kommunikation durch Interferenzen gestört wird. Ein Kostenvorteil durch den Kauf dieser alternativen Produkte wird rasch durch den Aufwand, um diese wunschgemäß einbinden und ansteuern zu können, kompensiert. Die Verfügbarkeit von technischen Datenblättern ist meistens dann auch eingeschränkt oder nur in wenigen Sprachen verfügbar.

Vor dem Anschließen von Sensoren an das Arduino Nano-Board sollte auch immer die Pinbelegung und die Pinbeschreibung laut Datenblatt des Herstellers studiert werden. Wenn ein Sensor falsch angeschlossen wird, stört das die Kommunikation zwischen Mikrocontroller und Rechner und die Messwerte können nicht oder nur fehlerhaft übertragen werden. Daher gilt auch hier das Motto, sich vor dem Programmieren und dem Anschließen des Boards mit der Hardware (mit den Datenblättern) zu beschäftigen um den Zeitaufwand für eine Fehlersuche und -korrektur (*Troubleshooting*) zu vermeiden.

Die Bereitschaft sich mit Virtual Reality und mit Virtual Reality-Brillen „einzulassen“ ist eine wesentliche Voraussetzung dafür, den besonderen immersiven Reiz, der Virtual Reality begleitet, zu verstehen. Für die Gamer-Community besteht ein Grundverständnis für diese Technologie. Im beruflichen Umfeld kann damit sehr viel an Innovation geleistet und an Wissensvermittlung bewirkt werden.

12.2 Ergebnisse

Zusammengefasst kann festgestellt werden, dass durch das Verständnis hinter der Logik und dem Algorithmus zur Anwendung unterschiedlicher Programmiersprachen für die Programmierung dieser seriellen Schnittstelle zum Auslesen von Messwerten und der Übertragung und Darstellung dieser in der Virtuellen Welt, es auch relativ schnell und einfach möglich ist, ähnliche Programmierhochsprachen anzuwenden. Eine besonders spannende Herausforderung war dabei, dass es zu dieser Form der seriellen Schnittstelle bisher keine wissenschaftlichen Arbeiten gibt.

Die Umsetzung der somit erstellten Programmcodes in Arduino IDE und Unity 3D 2018 für das Auslesen von Messwerten über ein Arduino kompatibles Board und das Ansteuern von LEDs über die virtuelle Welt sowie der Aufbau der virtuellen Umgebung (Einrichten und Koppeln der HTC Vive VR-Brille und der zwei

VR-Controller) konnte nach empirischen Versuchen und Tests erfolgreich umgesetzt und gemeinsam mit einem Prototyp eines LED-Boards der FH CAMPUS 02 präsentiert und übergeben werden.

Besonders interessant war es auch, die Virtual Reality-Technologien und den Aufbau der Virtual Reality-Brillen zu analysieren und damit einen Einblick zu bekommen, was das Eintauchen in die Virtuelle Welt für die Anwender so besonders macht.

12.3 Ausblick

Wenn in den Medien über Virtual Reality und deren Anwendungen berichtet wird, beschränkt sich die Berichterstattung schnell auf den Gaming-Bereich. In der ersten Betrachtung ist das nicht schlecht, allerdings sind die Bericht oft von einer pessimistischen Grundhaltung geprägt und zeigen wenig das große Potential für z.B. die Luft- und Raumfahrt, die Forschung und Medizin, die industriellen Anwendung oder für den Schulungs- und Ausbildungsbereich, die Virtual Reality mit sich bringt. Es wird daher Zeit, sich diesem Thema, auch medial zu näheren um die zahlreichen Ideen, Start-ups und Visionäre, die sich rund um dieses Thema gebildet haben, zu unterstützen und zu fördern.

Auch die Hersteller der VR-Brillen arbeiten an Verbesserungen zum effizienteren Einsatz. So begeistert die PIMAX 8K schon heute mit ihrem modularen Aufbau. Das ist ein Trend, dem auch andere Hersteller von Virtual Reality-Brillen zukünftig folgen werden. So können Brillenträger z.B. bei dieser VR-Brille maßgeschneiderte Linsen mit individueller Dioptrien-Zahl bestellen. In absehbarer Zeit werden zudem Eye-Tracking-Module, Wireless-Module (damit sollte auch der „Kabelsalat“ endgültig der Geschichte angehören), Handtracking-Module, Ventilator-Module und ein Geruchsmodul, welches in Zusammenarbeit mit einem japanischen Partner entwickelt wird, auch massentauglich. Diese Module werden zweifellos dazu beitragen, den industriellen Einsatz z.B. für Produktpräsentationen massiv zu forcieren.

Doch wie wird sich Virtual Reality in den nächsten 10 bis 20 Jahren entwickeln? Eine erste Prognose hierzu wagt Kevin Kelly, Mitbegründer und langjähriger Chefredakteur des renommierten Tech-Magazins Wired. In seinem Buch „*The Inevitable: Understanding the 12 technological forces that will shape our future*“ beschreibt er, wie VR-Technologie nach und nach in unser aller Leben Einzug hält - nicht zuletzt, weil auch Augmented Reality (AR) immer beliebter wird und für weitere technische Innovationen in den Sektoren AR und VR sorgen wird.

Als größte treibende Kraft hinter VR nennt Kelly das Verlangen der Menschen, in abwechslungsreichen VR-Umgebungen und eigens dafür entwickelten sozialen Netzwerken mit anderen zu kommunizieren, zu arbeiten und natürlich zu spielen. Gemäß einem Zitat von Kelly wird Virtual Reality in unmittelbarer Zukunft „das sozialste aller sozialen Netzwerke“.

LITERATURVERZEICHNIS

Gedruckte Werke

Schröter, Jens (Hrsg) (2012): *Echtzeit und Echtraum. Zur Medialität und Ästhetik von Augmented Reality Applikationen*, in AugenBlick, Marburger Hefte zur Medienwissenschaft Nr. 51, Marburg

Bartmann, Erik (Hrsg) (2017): *Mit Arduino die elektronische Welt entdecken*, Bombini Verlag, Bonn

Schalk, Gerhard H. (Hrsg) (2019): *Mikroprozessortechnik*, Version 0.8, Vorlesungsskript für die Fachhochschule CAMPUS 02, Graz

Korgel, Daniel (Hrsg) (2018): *Virtual Reality Spiele entwickeln mit Unity*, Carl Hanser Verlag, München

Slater, Mel, Wilbur, Sylvia (Hrsg.) (1997): *Presence: Virtual and Augmented Reality*, Volume 5, Issue 6, Massachusetts Institute of Technology

Dörner, Broll, Grimm, Jung (Hrsg) (2013): *Virtual und Augmented Reality (VR/AR)*, Springer Verlag, Berlin, Heidelberg

Niederlag, Wolfgang; Lemke, Heinz U.; Lehrach, Hans; Petgen, Heinz-Otto; Voelker, Wolfgang; Ertl Georg (Hrsg) (2014): *Der virtuelle Patient,* 2. Auflage, Walter de Gruyter Verlag, Berlin/Boston

Industriemagazin, „Das Österreichische Industrie Magazin“ (2019), *Spezialautomatisierung 5G*, WEKA Industrie Medien GmbH, Wien

Universitäts-Professor Dr. Dr. Fuchs, Thomas; Breyer, Thiemo (Hrsg.) (2013): *Grenzen der Empathie. Philosophische, psychologische und anthropologische Perspektiven (Übergänge)*, 1. Auflage, Wilhelm Fink Verlag, München

Kainka, Fabian (Hrsg) (2013): *Turn on your creativity, Das Franzis Starterpaket, Arduino Leonardo*, 1. Auflage, Franzis Verlag, Haar bei München

E-Books

Oliver, Thomas; Metzger, Dirk; Niegemann, Helmut; Welk, Markus; Becker, Thomas (Hrsg) (2018): *Digitalisierung in der Aus- und Weiterbildung: Virtual und Augmented Reality für Industrie 4.0*, Springer Verlag, Berlin/Heidelberg

SAMSUNG (Hrsg) (2015): *Benutzerhandbuch SM-322*, GH68-45048C Rev.1.0

Brühlmann, Thomas (Hrsg) (2015): *Arduino Praxiseinstieg*, 3. Auflage, mitp Verlags GmbH & Co. KG

Online-Quellen

VR-NERDS, A Virtual Reality Showcase, *Die Geschichte der virtuellen Realität*, https://www.vrnerds.de/die-geschichte-der-virtuellen-realitaet/ [Stand: 14.08.2019]

Virtual Reality Magazin (2019), *Virtuelle Lernumgebungen für Handwerk und Industrie*, https://www.virtual-reality-magazin.de/virtuelle-lernumgebungen-fuer-handwerk-und-industrie [Stand: 27.05.2019]

Absolventa GmbH (2019), Spezialistin für das Recruiting von Nachwuchstalenten: *XYZ – Generationen auf dem Arbeitsmarkt*, https://www.absolventa.de/karriereguide/berufseinsteiger-wissen/xyz-generationen-arbeitsmarkt-ueberblick [Stand: 27.05.2019]

OnlyVR (2015), Das VR Spiele, Entertainment und Technik Magazin: *Immersion und Präsenz*, http://www.onlyvr.de/virtual-reality/gefahren [Stand: 07.05.2019]

OnlyVR (2015), Das VR Spiele, Entertainment und Technik Magazin: *VR Gefahren und Risiken*, http://www.onlyvr.de/virtual-reality/immersion [Stand: 07.05.2019]

CONRAD (2017), *Was ist Virtual Reality und wie funktioniert VR?,* https://www.conrad.de/de/ratgeber/multimedia/virtual-reality.html [Stand: 28.08.2019]

PwC (2019), *Die Zukunft ist virtuell*, https://www.pwc.at/de/newsletter/digital/virtual-reality.html [Stand: 24.11.2019]

Thöing, Sebastian (2011), PC Games, *Der Motion Blur-Effekt in PC-Spielen: Für höhere Geschwindigkeiten und mehr Realismus*, https://www.pcgames.de/Panorama-Thema-233992/News/Der-Motion-Blur-Effekt-in-PC-Spielen-Fuer-hoehere-Geschwindigkeiten-und-mehr-Realismus-848965/ [Stand: 30.05.2019]

Donath, Andreas (2014), golem.de, IT-News für Profis: *Facebook kauft Oculus VR für 2 Milliarden US-Dollar*, https://www.golem.de/news/uebernahme-facebook-kauft-oculus-vr-fuer-2-milliarden-us-dollar-1403-105375.html [Stand: 25.05.2019]

OnlyVR (2015), Das VR Spiele, Entertainment und Technik Magazin: *VR-Brillen - Eine kleine Übersicht*, http://www.onlyvr.de/vr-brillen [Stand: 17.05.2019]

Biber, Michael Robert (2018), new direction: *VR-Brille - wie funktioniert sie und welche Brillen gibt es*, https://www.newdirection.de/blog/vr-brille-funktion-modelle [Stand: 17.05.2019]

Janssen, Jan-Keno (2019), heise online, c't Magazin: *VR-Headsets Pimax 5K+ und BE ausprobiert: Deutlich besser als die Prototypen*, https://www.heise.de/newsticker/meldung/VR-Headsets-Pimax-5K-und-BE-ausprobiert-Deutlich-besser-als-Prototypen-4268923.html [Stand: 22.05.2019]

Pertzborn, David Hrsg. (2019), ComputerBase: Virtual Reality: *Diese VR-Headsets erscheinen im Jahr 2019*, https://www.computerbase.de/2019-04/virtual-reality-diese-hardware-2019/ [Stand: 25.11.2019]

MIXED (2018), das Online-Magazin für Mixed Reality und die Zukunft der Computer: *StarVR: Neue positive Eindrücke zur Highend-VR-Brille*, https://mixed.de/starvr-erste-eindruecke-zur-neuen-highend-vr-brille/ [Stand: 22.05.2019]

Bezmalinovic, Tomislav (2018), MIXED, das Online-Magazin für Mixed Reality und die Zukunft der Computer: *Teuer, aber auch gut? Erstes Hands-On mit der 5K-Profi-Brille XTAL*, https://mixed.de/teuer-aber-auch-gut-erstes-hands-on-mit-der-5k-profi-brille-xtal/ [Stand: 22.05.2019]

Pro-physik.de, das Physikportal: *Neuer Sensor für eine elektronische Haut*, https://www.pro-physik.de/nachrichten/neuer-sensor-fuer-eine-elektronische-haut [Stand: 20.01.2020]

Jansen, Tom (Hrsg), Digital Production (2019), *Unreal Engine 4.22: Ein Mega-Update*, https://www.digitalproduction.com/2019/04/02/unreal-engine-4-22-ein-mega-update/ [Stand: 11.11.2019]

Viscircle GmbH (2018), *Unreal vs. Unity: Welche Software sollten Gamedeveloper wählen?*, https://viscircle.de/unreal-engine-vs-unity-welche-software-sollten-gamedeveloper-waehlen/ [Stand: 11.09.2019]

Snieders, Ralf (2019): *Anleitung für Arduino von Funduino GmbH*, https://funduino.de/anleitung-dht11-dht22 [Stand: 01.05.2019]

Atmel, *Atmel-8271JS-AVR- ATmega-Datasheet (2015),* https://www.mouser.at/datasheet/2/268/Atmel-8271-8-bit-AVR-Microcontroller-ATmega48A-48P-1315288.pdf [Stand: 01.05.2019]

Snieders, Ralf (2019): *Anleitung für Arduino von Funduino GmbH*, https://funduino.de/anleitung-dht11-dht22 [Stand: 01.05.2019]

Schnabel, Patrick (2017): *Elektronik Fibel*, https://www.elektronik-kompendium.de/sites/bau/0201111.htm [Stand: 13.05.2019]

Biagioli, Adrian (2016): *Virtual Button Designer for HTC Vive Controllers*, https://github.com/Flafla2/Vive-Virtual-Button-Designer/blob/master/README.md [Stand: 30.04.2019]

VRTK - Virtual Reality-Toolkit, powered by Readme (2018): *Welcome to VRTK*, https://vrtoolkit.readme.io/docs/summary [Stand: 03.06.2019]

C# Tutorial, *Grundlagen Datenarten*, https://sharp.net-tutorials.com/de/108/grundlagen/datenarten/ [Stand: 30.11.2019]

Feldbusse.de - Ihr Wegweiser für industrielle Netzwerke (2019): *Modbus-TCP – Architektur und Protokoll*, https://www.feldbusse.de/ModbusTCP/modbustcp_protokoll.shtml [Stand: 10.06.2019]

ABBILDUNGSVERZEICHNIS

Abb. 1: Virtual Reality Training, Quelle: eLearning Industrie (2018), Online-Quelle [26.05.2019]..... 2

Abb. 2: Global Virtual Reality Market, Quelle: Reuters Editorial News (2018), Online-Quelle [30.04.2019]. 3

Abb. 3: Gartner Hype Cycle for Emerging Technologies, Quelle: Gartner.com (2017), Online-Quelle [24.11.2019]..... 8

Abb. 4: Überblick über die Generationen, Quelle: Absolventa GmbH (2019), Online-Quelle [26.05.2019]. 9

Abb. 5: The Reality of Augmented Reality, Quelle: SD Times (2018), Online-Quelle [30.08.2019]. 11

Abb. 6: Unterschied zwischen Augmented und Virtual Reality, Quelle: Eigene Darstellung. 12

Abb. 7: Das EVR Studio VR Projekt *„Project M"-Teaser*, Quelle: MIXED (2017), Quelle: Eigene Darstellung. 14

Abb. 8: Big Data – Internet of Things (IoT), Quelle: SAS Best Pracitces (2018), Online-Quelle [19.01.2020]..... 17

Abb. 9: Display der Highend Virtual Reality-Brille StarVR, Quelle: MIXED (2018), Online-Quelle [22.05.2019]..... 22

Abb. 10: Stereoskopisches Sehen, Quelle: Conrad (2017), Online-Quelle [30.08.2019]. 23

Abb. 11: Bildqualität bei Virtual Reality-Brillen im Überblick, Quelle: VR-NERDS (2018), Online-Quelle [22.05.2019]..... 27

Abb. 12: Bildqualität bei Virtual Reality-Brillen im Überblick, Quelle: heise online (2019), Online-Quelle [22.05.2019]..... 31

Abb. 13: Deutliche breiteres Sichtfeld der Pimax VR-Brille als die Oculus Rift oder die HTC Vive VR-Brille, Quelle: heise online (2019), Online-Quelle [22.05.2019]. 32

Abb. 14: Strahlenbüschel (God-Rays) im Kloster Maulbronn, Quelle: fotopraxis (2015), Online-Quelle [25.11.2019]..... 32

Abb. 15: Das Leap-Motion Modul der VR-Brille XTAL, Quelle: MIXED (2018), Online-Quelle [22.05.2019]. 34

Abb. 16: Der HoloSuit Ganzkörperanzug, Quelle: HoloSuit (2018), Online-Quelle [27.05.2019]. 35

Abb. 17: Der HoloSuit Ganzkörperanzug, Quelle: HoloSuit (2018), Online-Quelle [27.05.2019]. 36

Abb. 18: HoloSuit Bewegungssensoren, Quelle: VRPLAYGROUND (2018), Online-Quelle [27.05.2019]. 37

Abb. 19: HoloSuit – haptisches Feedback, Quelle: VRPLAYGROUND (2018), Online-Quelle [28.05.2019]. 37

Abb. 20: Unity Logo, Quelle: Unity Technologies (2019), Online-Quelle [25.05.2019]. 39

Abb. 21: Unreal Logo, Quelle: unrealengine.com (2019), Online-Quelle [18.11.2019]. 40

Abb. 22: Unreal Logo, Quelle: Epic Games (2019), Online-Quelle [10.09.2019]..... 40

Abb. 23 Darstellung von Reflektionen zur Simulation von akkuraten Umgebungs-Reflektionen auf reflektierenden Flächen in der Unreal Engine, Quelle: Digital Production (2019), Online-Quelle [18.11.2019]..... 41

Abb. 24: Grober Entwurf eines Blackbox-Modelles und Groblayout für den Entwurf in Hard- und Software, Quelle: Eigene Darstellung..... 44

Abb. 25: Arduino Nano-Board V3.0, Quelle: core Electronics (2019), Online-Quelle [01.05.2019]..... 46

Abb. 26: Pinbelegung des Arduino Nano-Board V3.0 für den DHT11 Sensor, Quelle: Eigene Darstellung. 47

Abb. 27: Pinbelegung des Arduino Nano-Board V3.0, Quelle: CIRCUITS TODAY (2018), Online-Quelle [01.05.2019]..... 47

Abb. 28: Pinbeschreibung des Arduino Nano-Board V3.0, Quelle: The engineering projects (2018), Online-Quelle [01.05.2019]..... 48

Abb. 29: Pinbeschreibung des Arduino Nano-Board V3.0, Quelle: CIRCUITS TODAY (2018), Online-Quelle [01.05.2019]. 48

Abb. 30: Digitaler Temperatur- und Luftfeuchtigkeitssensor DHT11, Quelle: Eigene Darstellung. 52

Abb. 31: Datenformat der Übertragung des Sensors DHT11, Quelle: Eigene Darstellung. 54

Abb. 32: Berechnungsbeispiel für 40 über den Sensor erhaltene Bit-Daten, Quelle: Eigene Darstellung. 54

Abb. 33: EAGLE Logo, Quelle: AUTODESK (2019), Online-Quelle [30.04.2019]..... 55

Abb. 34: LED-Board-Leiterplattenlayout, Quelle: Eigene Darstellung. 56

Abb. 35: LED-Board-3D-Leiterplattmodell, Quelle: FH CAMPUS 02, [18.11.2019]..... 56

Abb. 36:LED- Schaltungstechnik Quelle: Elektronikwissen (2019), Online-Quelle [13.05.2019]. 57

Abb. 37: Farbcodierung des Vorwiderstandes, Quelle: digikey (2019), Online-Quelle [24.05.2019]..... 59

Abb. 38: Auf das Arduino kompatible Board aufgesetzte selbstdesignten Leiterplatte mit 4 LEDs und 220 Ohm Vorwiderständen, Quelle: Eigene Darstellung..... 60

Abb. 39: LED-Board mit einigen angesteuerten LEDs, Quelle: Eigene Darstellung..... 61

Abb. 40: Die HTC Vive VR-Hardware, Quelle: IT-Kurzzeitvermietung (2018), Online-Quelle [30.05.2019] 62

Abb. 41: Die HTC Vive VR-Brille, Quelle: Conrad Electronic (2018), Online-Quelle [30.05.2019] 62

Abb. 42: Vorderseite und Seitenansicht der HTC Vive Virtual Reality-Brille, Quelle: HTC Vive Handbuch (2018), Online-Quelle [04.06.2019] 63

Abb. 43: HTC VIVE Controller, Quelle: STEAM Community (2018), Online-Quelle [30.04.2019]..... 64

Abb. 44: Navigation der VR-Controller, Quelle: HTC Vive Handbuch (2018), Online-Quelle [04.06.2019]64

Abb. 45: Einrichtung der HTC Basis-Stationen, Quelle: Vive (2019), Online-Quelle [30.05.2019]..... 65

Abb. 46: Arduino Logo, Quelle: Arduino (2019), Online-Quelle [30.04.2019]. 66

Abb. 47: Arduino Entwicklungsumgebung IDE, Quelle: Eigene Darstellung. 67

Abb. 48: Pinbelegung des LED- und Arduino kompatiblen-Boards, Quelle: Eigene Darstellung. 68

Abb. 49: Struktur der C-Sprache in Arduino IDE Quelle: Arduino Programmier-handbuch (2018), Online-Quelle [03.06.2019]. 70

Abb. 50: 32 Schlüsselwörter der Sprache C/C++, Quelle: LEARN C (2019), Online-Quelle [03.06.2019].70

Abb. 51: Starten des Arduino Library Managers, Quelle: Eigene Darstellung. 72

Abb. 52: Arduino Library Manager, Quelle: Eigene Darstellung. 72

Abb. 53: Ansicht des Unity 3D 2018-Editors, Quelle: Eigene Darstellung. 76

Abb. 54: Ansicht des Unity 3D 2018-Editors, Quelle: Eigene Darstellung. 78

Abb. 55: *Inspector*-Bereich in Unity 3D 2018-Editors, Quelle: Eigene Darstellung..... 78

Abb. 56: Transform-Tools in Unity 3D 2018-Editors, Quelle: Eigene Darstellung. 79

Abb. 57: Project Browser in Unity 3D 2018-Editors, Quelle: Eigene Darstellung. 81

Abb. 58: Die Virtual Reality-SDKs unter Unity 3D 2018, Quelle: Eigene Darstellung. 81

Abb. 59: Bewegliche Skalen des analogen Thermometers und des analogen Hygrometers, Quelle: Eigene Darstellung. 83

Abb. 60: Virtual Reality Toolkit für die Unity Engine, Quelle: VRTK (2018), Online-Quelle [03.06.2019]. 84

Abb. 61: Auszug aus dem C#-script in Unity 3D 2018, Quelle: Eigene Darstellung. 89

Abb. 62: Messtisch für die virtuelle Umgebung, Quelle: Eigene Darstellung. 90

Abb. 63: Messtisch in der virtuellen Umgebung in Unity 3D 2018, Quelle: Eigene Darstellung. 92

Abb. 64: Messtisch in der virtuellen Umgebung in Unity 3D 2018, Quelle: Eigene Darstellung. 92

Abb. 65: Einbindung der Anwendung EasyModbus in Unity, Quelle: Eigene Darstellung..... 94

Abb. 66: Easy Modbus Server Simulator, Quelle: Eigene Darstellung. 94

Abb. 67: Easy Modbus Server Simulator – Schnittstelle in Unity, Quelle: Eigene Darstellung..... 95

Abb. 68: Easy Modbus Server Simulator – Holding Registers, Quelle: Eigene Darstellung..... 95

ABKÜRZUNGSVERZEICHNIS / GLOSSAR

Die Kurzbeschreibung der einzelnen Abkürzungen wurde unterschiedlichen Internetquellen entnommen und soll zum allgemeinen besseren Verständnis beitragen.

ADC	*Analog-to-digital converter* (Analog-Digital-Wandler oder A/D-Wandler)
AR	Augmented Reality (Unter „Erweiterter Realität“ versteht man die computergestützte Erweiterung der Realitätswahrnehmung.)
CES	International Consumer Electronics Show (Diese Show ist eine der weltweit größten Fachmessen für Unterhaltungselektronik. Sie findet jährlich im Jänner im Las Vegas Convention Center (LVCC) in Las Vegas statt.)
COM	*Component Object Model* (Darunter versteht man ein *Framework* zur Erstellung von Softwarekomponenten.)
CPU	*Central processing unit* (Zentrale Rechen- und Steuereinheit eines Computers)
CRC	*Cyclic redundancy check* (Die zyklische Redundanzprüfung ist ein Verfahren zur Bestimmung eines Prüfwerts für Daten, um Fehler bei der Übertragung oder Speicherung erkennen zu können.)
EEPROM	*Electrically Erasable Programmable Read-Only Memor* (Nichtflüchtiger Speicher, der einmal gespeicherte Daten auch nach dem Verlust der Versorgungsspannung behält.)
FoV	*Field of view* (Das Sichtfeld ist die Größe des Bereiches, der in Abhängigkeit von den Linsen und dem Display für den Anwender einer VR-Brille sichtbar ist. Je größer das FoV, desto besser ist die Immersion.)
FTDI	*Future Technology Devices International* (USB-RS232-Interface-Chips der Firma Future Technology Devices International)
GND	Ground (Darunter versteht man einen gemeinsamen Signal- und Stromrückleitungspfad.)
GPIO	*General Purpose Input Output*
HMD	*Head-Mounted Display* (Ist ein auf dem Kopf zu tragendes visuelles Ausgabegerät. Es präsentiert Bilder entweder auf einem augennahen Bildschirm oder projiziert sie direkt auf die Netzhaut.)
HTML	*Hypertext Markup Language* (Daraunter versteht man eine textbasierte Auszeichnungssprache zur Strukturierung elektronischer Dokumente wie Texte mit *Hyperlinks*, Bildern und anderen Inhalten.)
ID	Ein Identifikator ist ein mit einer bestimmten Identität verknüpftes Merkmal zur eindeutigen Identifizierung des tragenden Objekts.
I²C	*Inter-Integrated Circuit* (Der Inter IC-Bus wurde von Philipps entwickelt. Er ist ein Bus, der die Kommunikation zwischen verschiedenen integrierten Schaltungen ermöglicht bzw. bereitstellen soll.)

ICSP *In-circuit serial programming* (ICSP oder auch *in-system*-Programmierung (ISP) genannt, ermöglicht das Programmieren einer logischen Schaltung direkt im Einsatzsystem. Dazu wird meist eine einfache serielle Verbindung genutzt, z.B. JTAG oder SPI. Der Vorteil der *In-System*-Programmierung ist, dass der zu programmierende Schaltkreis nicht mehr aus dem Zielsystem entfernt werden muss. Die Pins des jeweilig verwendeten Programmes sind auf Stiftleisten aufgelegt und können mittels *Jumpwires* mit dem jeweiligen Pin des Mikrocontrollers verbunden werden. Für Mikrocontroller der Firma Atmel ist ein standardisierter 6-poliger ICSP Stecker vorhanden.)

IIoT *Industrial Internet of Things* (Das “Internet der Dinge” stellt ein Konzept im produzierenden Umfeld für industrielle Prozesse und Abläufe dar. Ziele des IIoT sind Kostensenkungen in der Produktion, schnellere und effizientere Prozesse und die Realisierung neuer Geschäftsmodelle. Produktionsprozesse lassen sich dabei mittels der erhobenen, analysierten und verarbeiteten Daten automatisieren und flexibel an sich permanent verändernde Einflussfaktoren anpassen.)

IPD Interpupillardistanz (Als IPD wird in der Augenheilkunde und Augenoptik der Augenabstand bezeichnet. Darunter versteht man den in mm angegebenen Abstand beider Augen zueinander. Er ist beim Anpassen einer Brille zu berücksichtigen und dient der Ausrichtung der optischen Achsen von Brillengläsern (Hauptdurchblickspunkt) an denen der Augen bzw. Augenachsen.)

ISP *In-System-Programmer* (siehe dazu auch ICSP)

LCD *Liquid Crystal Display* (Darunter versteht man die Verwendung von Flüssigkristallen, deren Ausrichtung durch elektrische Impulse gesteuert werden. Je nach Ausrichtung lassen die Kristalle unterschiedlich viel Licht durch, so ergeben sich die unterschiedlichen Farben. Die Grundvoraussetzung dafür ist eine Hintergrundbeleuchtung.)

LED *Light Emitting Diode* (Lichterzeugende Halbleiterdiode)

OCIO *OpenColorIO* (Dieses System wurde speziell für *Workflows* mit vollständigem Farb-management entwickelt. *OpenColorIO* wird zwar hauptsächlich in Filmproduktionen verwendet, lässt sich aber für jede Situation einsetzen, in der es von Anfang bis Ende auf ein präzises Farbmanagement ankommt.)

OLED *Organic Light Emitting Diode* (OLED Displays bestehen aus LEDs aus organischen Halbleitermaterialien, die selbst so stark leuchten, dass auf eine Hintergrundbeleuchtung verzichtet werden kann. Die OLED-Technologie vereint damit die Vorteile von LCD und Plasma. Neben einem breiteren Farbspektrum, können mit OLED Displays eine größere Helligkeit und gleichzeitig bessere Schwarzwerte erzielt werden.)

PWM *Pulse Width Modulation* (Pulsweitenmodulation ist grundsätzlich eine Modulationsart zum Erzeugen eines analogen Signals anhand einer digitalen Signalquelle. Arduino verwendet die PWM-Technik zur Steuerung analoger Schaltungen mit seinen digitalen Ausgängen. Die digitale Steuerung wird nur im binären Format eingeschaltet (volle 5 V) oder ausgeschaltet (0 V), und dieses Ein/Aus-Muster erzeugt ein Rechteckwellensignal. Wenn

	man beispielsweise eine halbhelle LED wünscht, kann man entweder den Strom über die LED halbieren oder die flexiblere PWM-Technik verwenden, indem man ein Rechtecksignal mit 50 % (Farbverhältnis) an die LED sendet. Dieses wird durch das logarithmische Sehen des Auges aber nicht halb so hell sein.)
RAM	*Random Access Memory*
RGB	(Ein Rot-Grün-Blau-Farbraum ist ein additiver Farbraum, der Farbwahrnehmungen durch das additive Mischen der drei Grundfarben Rot, Grün und Blau nachbildet.)
RX0	*Received eXchange Data*
SFR	*Special Function Register*
SDK	Software Development Kit (Darunter versteht man eine Sammlung von Programmierwerkzeugen und Programmbibliotheken, die zur Entwicklung von Software dienen. In der Regel gibt es zu jeder Programmiersprache ein *Software Development Kit*. Eine besondere Rolle nehmen SDKs für Betriebssysteme ein. Sie enthalten die notwendigen *Compiler*, Dienstprogramme und Informationen, um für dieses überhaupt Software zu entwickeln.)
SPI	*Serial Peripheral Interface* (Darunter versteht man einen synchronen und seriellen Bus, der nach dem *Master-Slave*-Prinzip arbeitet und der von der Firma Motorola entwickelt wurde.)
TCP	*Transmission Control Protocol/Internet Protocol*
TTL	*Time to live* (Gültigkeitsdauer der Daten im Rechnernetz)
TX1	*Transmit eXchange Data*
USART	*Universal Synchronous/Asynchronous Receiver Transmitter* (USART (USARTs bieten die zusätzliche Möglichkeit einer synchronen Datenübertragung unter Verwendung einer gemeinsame Taktleitung, die dem Empfänger den genauen Zeitpunkt nennt, wann die Datenleitung den nächsten gültigen Pegel eingenommen hat. Die Taktleitung synchronisiert Sender und Empfänger.)
USB	*Universal Serial Bus* (Darunter versteht man einen seriellen Standardanschluss. Die Transferrate hängt von der jeweiligen USB-Generation ab.)
VCC	*Voltage* (Spannungsbezeichnung)
VR	Kurzform für Virtual Reality (Darstellung und gleichzeitige Wahrnehmung einer in Echtzeit computergenerierten, interaktiven virtuellen Umgebung.)
VRTK	*Virtual Reality Toolkit* unter Unity 3D 2018.